SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2104

TEHNIKE UČENJA VIŠESTRUKOSTI ZA POVEĆANJE UČINKOVITOSTI ANALIZE KOJA KORISTI SPOREDNA SVOJSTVA KRIPTOGRAFSKIH UREĐAJA

Eugen Vušak

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2104

TEHNIKE UČENJA VIŠESTRUKOSTI ZA POVEĆANJE UČINKOVITOSTI ANALIZE KOJA KORISTI SPOREDNA SVOJSTVA KRIPTOGRAFSKIH UREĐAJA

Eugen Vušak

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 13. ožujka 2020.

DIPLOMSKI ZADATAK br. 2104

Pristupnik:	**Eugen Vušak (0036493959)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Tehnike učenja višestrukosti za povećanje učinkovitosti analize koja koristi sporedna svojstva kriptografskih uređaja**

Opis zadatka:

U ovom diplomskom radu razmatra se primjena postupaka učenja višestrukosti (engl. manifold learning) za povećanje učinkovitosti analize koja koristi sporedna svojstva kriptografskih uređaja (engl. side-channel analysis, dalje: SCA). U radu je potrebno proučiti i opisati postupke analize SCA, njegovo podrijetlo i raspon primjene. Također, potrebno je proučiti tehnike učenja višestrukosti u kontekstu različitih pristupa koji postoje za smanjenje dimenzionalnosti skupa podataka. Zadatak u diplomskom radu bit će primijeniti postupke učenja višestrukosti na nekoliko skupova podataka koji sadrže mjerenja sporednih svojstava uređaja, a koji su dostupni istraživačima u ovom području. Procjena uspješnosti primjene postupaka učenja višestrukosti temeljit će se na rezultatima klasifikacijskih modela (npr. slučajna šuma, stroj s potpornim vektorima) i na nekoliko dodatnih mjera uspješnosti detekcije tajnog ključa kao što su stopa uspješnosti i entropija pogađanja. U radu je potrebno prikazati i usporedbu uspješnosti postupaka učenja višestrukosti s tradicionalnim postupcima za smanjenje dimenzionalnosti kao što je analiza glavnih komponenti. Cilj rada je pokazati do kojeg stupnja je moguće smanjiti skup značajki u skupovima podataka a da se još uvijek u prosjeku očekivano uspješno može provesti analiza SCA.

Rok za predaju rada: 30. lipnja 2020.

Ovim putem

htio bih se zahvaliti Sveučilištu u Zagrebu i Fakultetu elektrotehnike i računarstva na pruženoj prilici za stjecanje višeg obrazovanja. Uz to htio bih se dodatno zahvaliti svim profesorima i asistentima koji svojim znanjem i trudom to obrazovanje pokretali. Posebice bi se htio zahvaliti svojem mentoru, izv. prof. dr. sc. Alanu Joviću, na strpljenju i korisnim savjetima, ne samo za vrijeme pisanja diplomskog rada, već i na mnogim drugim projektima koji su prethodili ovaj rad. Konačno htio bih se zahvaliti svojoj obitelji, djevojci Mihaeli i prijatelju Marijanu na njihovoj nemjerljivoj podršci i pomoći.

SADRŽAJ

1. Uvod

U današnje vrijeme sve je više informacija zapisano u digitalnom obliku, zbog čega je njihova sigurnost postala bitnija nego ikada. Iz istog je razloga nastao značajan razvoj u području kriptografije, primjer čega su kriptografski algoritmi poput AES-a i kriptografski sustav RSA. Takvi moderni algoritmi vrlo su robusni zbog svoje snažne matematičke osnove čime se znatno otežavaju tradicionalni kriptografski napadi. Kriptografski algoritmi uglavnom su dizajnirani na vrlo apstraktnoj razini odvojenoj od fizičke implementacije na kojoj će se isti ti algoritmi izvoditi. Dakle, sa strane algoritma nije bitno hoće li se on izvoditi na generičnom procesoru, kakav možemo naći u osobnim računalima, ili pak na specijaliziranom sklopovlju izvedenom u obliku FPGA integriranog kruga ili ASIC-a. Drugim riječima, na kriptografski algoritam možemo gledati kao na "crnu kutiju" u koju ulazi originalna poruka (i dodatni parametri), a izlazi kriptirana poruka. Takav teoretski pogled ima tradicionalna kriptoanaliza te je on neosporivo bitan.

Međutim, stvarni svijet nije "crno-bijeli" te kriptografski algoritmi moraju biti implementirani na stvarnom, fizičkom uređaju koji je također ključan dio cijelog sustava. No, takav uređaj nije ni približno savršen kao njegov teorijski opis te će zbog svojih fizičkih svojstava uvijek emitirati dio informacija u okolinu. Dakle, za sustav možemo reći da se više ponaša kao tzv. "siva kutija", gdje je dio informacija poznat promatraču, a one se mogu manifestirati u brojnim oblicima, poput svijetla emitiranog iz LED dioda na uređaju, topline ili zvuka. Upravo takva, sporedna svojstva praktične implementacije kriptografskih algoritama, koristi područje kriptoanalize pod nazivom analiza sporednih kanala, (engl. *Side-Channel Analysis*, dalje SCA). Gotovo svi poznati napadi na kriptografske uređaje su napadi koji iskorištavaju njihova sporedna svojstva. Jedni od najpoznatijih takvih napada, zasigurno, su napadi "Meltdown" (Lipp et al., 2018) i "Spectre" (Kocher et al., 2019) koji su koristili mane u dizajnu modernih procesora kako bi dohvatili informacije za koje se vjerovalo da su potpuno zaštićene. Protumjere su morale biti napisane direktno u mikrokod procesora čime su degradirane performanse procesora, a tek je na kasnijim modelima uvedena promjena u samom dizajnu

procesora.

Sigurnost informacija očito je vrlo važna te kako bismo informacije zaštitili potrebno je istraživati nove metode analize. Naime, jedini način da popunimo sigurnosnu "rupu" je ako znamo da ona postoji. Ovaj diplomski rad temeljen je na radu od Picek et al. (2019), na koji će se nadovezati razmatranjem primjena tehnika učenja mnogostrukosti (engl. *Manifold Learning*) s ciljem povećanja učinkovitosti SCA smanjenjem skupa značajki, tj. dimenzionalnosti, no zadržavanjem većine informacija. Učenje višestrukosti jedan je od načina smanjenja dimenzionalnosti koji u obzir uzima topologiju podataka te pokušava smanjiti njihovu dimenzionalnost tako da se sačuvaju topološka svojstva originalnog prostora podataka.

Ovaj rad podijeljen je na sljedeći način. U prvom poglavlju bit će opisana generalna problematika, definiran cilj te obrazložena struktura rada. Drugo i treće poglavlje dat će teorijski uvod u dva velika područja na kojima počiva ovaj rad, a oni su, redom, analiza koja koristi sporedna svojstva uređaja i smanjenje dimenzionalnosti.

Drugo poglavlje bavit će se tematikom analize koja koristi sporedna sredstva, bit će opisani osnovni koncepti ovog područja kriptografije i načini klasifikacije napada. Uz to, u ovom poglavlju će biti pobrojani i detaljno opisani dosad poznati napadi, a konačno, na kraju i razne protumjere.

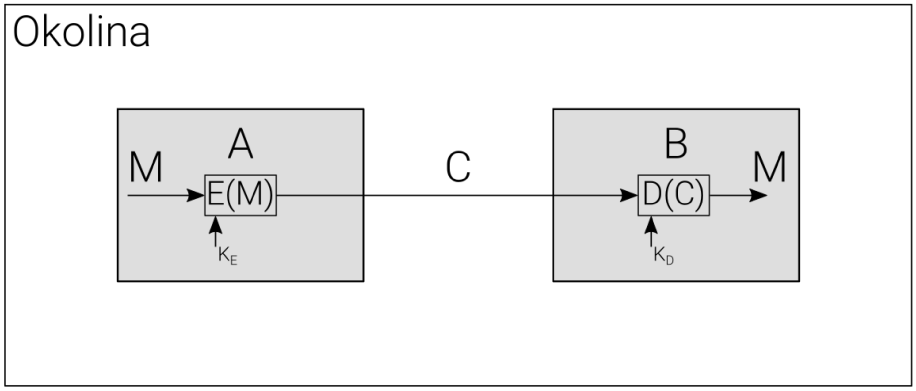
U trećem poglavlju, osim koncepta smanjenja dimenzionalnosti bit će opisane pripadne dvije velike metode; linearna i nelinearna. Za svaku će se ponaosob opisati temeljni način rada, a zatim i razne metode korištene u okviru njih. Na kraju poglavlja bit će prezentirana detaljna usporedba algoritama za smanjenje dimenzionalnosti na četiri primjera raznih priroda.

Zatim će u četvrtom poglavlju biti opisana primjena metoda smanjenja dimenzionalnosti u SCA, dat će se dodatan opis problema potreban za razumijevanje metode, bit će opisana metodologija dobivanja rezultata. Ti rezultati će, zatim, biti izloženi i diskutirani nakon čega će biti donesen zaključak.

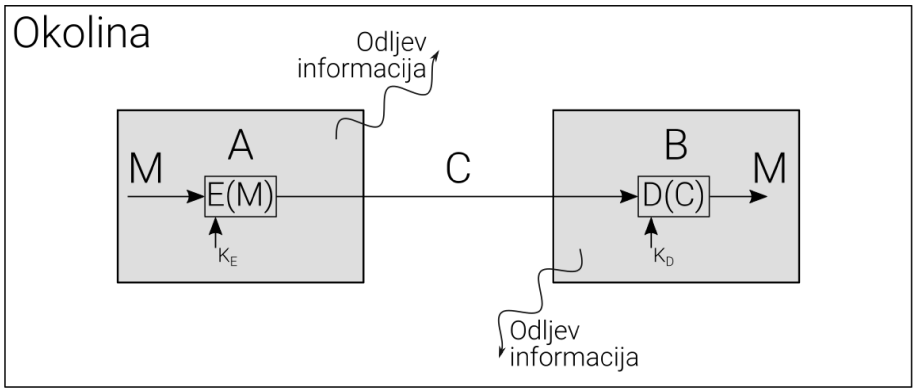
2. Analiza koja koristi sporedna svojstva uređaja

Kao što je već spomenuto, tradicionalno, u kriptoanalizi napadač (engl. *adversary*) zna samo kriptirane poruke, ako je riječ o napadu s kriptiranim tekstom (engl. *ciphertext attack*), ili originalne i kriptirane poruke, ako govorimo o napadu s čistim tekstom (engl. *plaintext attack*) te tako pokušava odgonetnuti nepoznate kriptirane poruke (slika 2.1). Takav pristup omogućio je, i diskutabilno je nužan, za ogroman pomak u kriptografiji te razvoj kriptografskih primitiva poput jednosmjernih hash funkcija, simetričnih i nesimetričnih kriptografskih algoritama i digitalnih potpisa. One su sastavni dio svih kriptosustava kao što su sigurni komunikacijski protokoli, RSA i novorastuće kriptovalute čiji primjer je vrlo popularan Bitcoin.

Ali, zbog fizičke prirode implementacije tih algoritama na elektroničkim uređajima postoji dodatni izvor informacija zanemaren u tradicionalnoj kriptoanalizi. Naime, svaki elektronički uređaj ima neku vrstu odljeva informacija, koji se može manifestirati u raznim oblicima od elektromagnetskog zračenja poput svjetla ili električne energije potrebne za njegov rad do trajanja izvođenja ili zvuka. Takve fizičke interakcije mogu biti prisluškivane od strane napadača te dati uvid u rad uređaja, a time i korisne informacije za provođenje napada, što je ilustrirano na slici 2.2. Sigurnosni sustav siguran je koliko i njegov najslabiji dio. Zanimljiva analogija za sagledati je kuća koja ima najsofisticiranija protuprovalna vrata, ali ima otvoren prozor odmah pored. Taj prozor je analogan informacijama kod kojih je došlo do odljeva neželjenim putem i njih nazivamo *sporednim svojstvima uređaja* (engl. *side-channel*) dok je područje koja analizira njihove eksploatacije upravo SCA. Lagano je uočiti da, za razliku od tradicionalne kriptoanalize koja kriptosustave gleda kao samo matematičke tvorevine, SCA se fokusira na fundamentalnu implementaciju istih te se iz tog razloga često u literaturi može vidjeti da se napadi SCA nazivaju i *implementacijski napadi* (engl. *implementation attacks*). To je ujedno i nedostatak ovakvih napada, jer za razliku od vrlo generičnih napada kriptoanalizom, ovakav napad cilja jedan ili malu skupinu uređaja.



Slika 2.1: Model tradicionalne kriptoanalize



Slika 2.2: Model SCA

Prema Zhou i Feng (2005) jedna od najranijih instanci napada SCA dogodila se još 1965. godine, kada je britanska tajna služba, MI6, pokušala probiti šifru koju je u to doba koristila egipatska ambasada, no slabi računalni resursi toga doba onemogućili su sve njihove pokušaje. Kako bi premostio taj problem, tadašnji znanstvenik GCHQ-a, P. Wright, predložio je da mikrofon bude postavljen blizu rotora kriptološkog uređaja korištenog od strane Egipćana i tako se omogućilo prisluškivanje zvukova kliktanja koje je uređaj proizvodio. Slušajući zvukove resetiranja rotora svako jutro britanska agencija je uspješno deducirala pozicije dva od tri rotora i time značajno smanjila računalne zahtjeve potrebne za dekripciju. MI6 je zbog ovog pothvata mogla prisluškivati komunikaciju ambasade godinama.

Moderne tehnike SCA je uveo Paul Kocher (1996; 1999; 2004), koji je napisao da budući da na količinu električne energije uređaja utječu podaci koji se obrađuju, mjerenja potrošnje energije sadrže podatke o kalkulacijama kruga (Kocher et al., 2011). Metode napada SCA pokazale su se znatno efektivnijima od konvencionalnih napada baziranih na matematičkoj analizi. Prilikom dizajniranja sigurnosnog sustava, dizajner na raspolaganju ima plato čvrstih formalnih metoda verifikacije kojima se u obzir

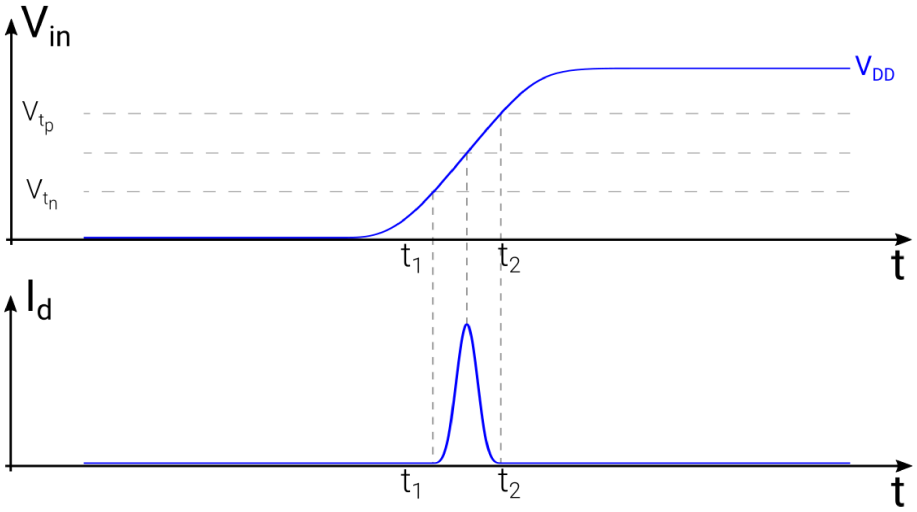
uzimaju različiti tipovi napada pomoću kojih onda razmatra svojstva sigurnosti. Međutim, situacija je puno teža kada se odmaknemo od idealiziranih matematičkih modela do implementacije u stvarnom, fizičkom, svijetu. U ovakvoj situaciji postavlja se mnoštvo pitanja, poput koliki pristup uređaju će napadač imati, na koji način može manipulirati ulazima, kakve izlaze može identificirati i razaznati. Ali, kao što smo rekli takvih izlaza ima puno te ih je teško definirati. Kako bi se dizajnirao siguran sustav potrebno je puno iskustva, a za testiranje nije moguće koristiti formalne metode, već su potrebne simulacije i mjerenja na stvarnom uređaju nerijetko koristeći skupocjenu opremu. Zbog velike raznolikosti izvora informacija te nemogućnosti standardizacije, najteži dio dizajniranja sigurnog sustava je upravo detektiranje nedostataka. No, "Historia est Magistra Vitae", povijest je učiteljica života te je za bolje razumijevanje važno razumjeti postojeće napade koji će biti bolje objašnjeni u odjeljku 2.3

2.1. Osnove napada SCA

Napadi SCA usko su vezani za postojanje fizički primjetnih fenomena nastalih zbog izvođenja programa na današnjim mikrokontrolerskim uređajima. Naprimjer, mikroprocesori koriste električnu snagu i vrijeme kako bi izveli zadani zadatak, a pritom, zbog nesavršene efikasnosti uređaja emitiraju elektromagnetsko (skrać. EM) zračenje jedno od kojih je toplina (Standaert, 2010). Iako postoji mnogo različitih izvora informacija koje se odlijevaju u okolinu, neke od kojih potencijalno još niti ne znamo, u ovom poglavlju, kako bi se ilustrirale tehnike SCA i dio pozadine od kuda one potiču, fokus će biti na dva tipa sporednih sredstava, a to su potrošnja energije i EM zračenje unutar CMOS tranzistora zbog njihove dominacije u svijetu integriranih krugova.

2.1.1. Potrošnja energije CMOS uređaja

Većina digitalnih uređaja zasniva se na CMOS vratima. To je sustav koji sa sastoji od dvije vrste tranzistora: pMOS tranzistora koji je spojen u načinu rada za pritezanje na napon napajanja (engl. *pull-up*) i nMOS tranzistora koji je spojen u načinu pritezanja na uzemljenje (engl. *pull-down*). Tako je moguće izgraditi osnovne digitalne krugove, a zatim pomoću njih i kompleksnije sustave. Statička CMOS vrata imaju tri izrazita izvora disipacije. Prvi nastaje zbog unutarnjih struja samog tranzistora. Drugi izvor rezultat je činjenice da postoji kratak period prilikom promjene vrata kada i pMOS i nMOS tranzistori vode u isto vrijeme, zbog kojeg dolazi do takozvanih struja kratkog spoja I_d vidljivih na slici 2.3.



Slika 2.3: Struja kratkog spoja prilikom promjene stanja

Treći razlog je zbog kapaciteta C_L tranzistora. Naime, tranzistori imaju elektrode odvojene s tankim slojem dielektričnog materijala i time zapravo jesu fundamentalno i kondenzatori. Zbog toga postoji određena *dinamička potrošnja energije* koja nastaje zbog punjenja i pražnjenja kondenzatora. Ona je posebno bitna u SCA jer je određena jednostavnom relacijom između unutarnjeg stanja uređaja i mjerljive fizikalne pojave. Taj odnos može biti izražen kao:

$$P_d = C_L V_{DD}^2 P(0 \rightarrow 1) f \tag{2.1}$$

gdje je $P(0 \rightarrow 1)$ vjerojatnost da se dogodi promjena $0 \rightarrow 1$, f radna frekvencija uređaja, a V_{DD} napon napajanja. Dakle, kada mjerimo potrošnju energije šiljci će se pojavljivati prilikom pražnjenja i punjenja kondenzatora, točnije prilikom promjene stanja iz 1 u 0 ili obrnuto. Upravo ovakve povezanosti su izvor odljeva informacija.

2.1.2. Elektromagnetsko zračenje CMOS uređaja

Kao što je u prošlom odjeljku pokazan odnos između podataka i potrošnje energije, moguće je pokazati odnos između podataka i EM zračenja. Postoje dvije bitne povezanosti, prva je da je EM zračenje ovisno o jakosti struje koja je povezana s podacima, a druga orijentacija EM polja je ovisna o smjeru kretanja struje, za koju je također očito da je ovisna o podacima. Takvo EM zračenje često se teorijski modelira pomoću Biot-Savartovog zakona:

$$dB = \frac{\mu I dl \times \hat{r}}{4\pi r^2} \tag{2.2}$$

Takvo zračenje također može biti opaženo mjerenjima te je neželjeni izvor informacija.

2.1.3. Modeli odljeva

Uzevši implementacijske detalje, poput onih opisanih u prošla dva poglavlja, razvijeni su brojni modeli odljeva koji se mogu koristiti za simuliranje napada ili povećavanje efikasnosti napada. Dva značajnija modela su model Hammingove udaljenosti i model Hammingove težine. Model Hammingove težine pretpostavlja da kada se računa vrijednost x_0 tada je odljev u korelaciji s Hammingovom težinom te vrijednosti, $H_W(x_0)$, dok model Hammingove udaljenosti pretpostavlja da kada u uređaju dođe do promjene iz vrijednosti x_0 u vrijednost x_1 tada vrijednost odljeva je u korelaciji s Hammingovom udaljenosti tih vrijednosti, tj. $H_D(x_0, x_1) = H_W(x_0 \oplus x_1)$. Oba modela kao pretpostavku uzimaju da događaji $0 \rightarrow 1$ i $1 \rightarrow 0$ utječu jednako na cjelokupnu potrošnju energije. Postoje kompleksniji modeli koji te pretpostavke umanjuju, primjerice koristeći različite odljeve za spomenute događaje ili dodjeljujući težinske vrijednosti doprinosima odljeva različitih dijelova sustava.

2.1.4. Mjerenje

Sa strane praktične implementacije napada SCA, dobro definirana mjerna shema je od primarne važnosti. Cilj je pretvoriti fizikalna svojstva ciljanog uređaja u digitalno iskoristive podatke. Elementi koji sačinjavaju takvu mjernu shemu generalno su:

- ciljani uređaj, npr. integrirani ili FPGA krug koji izvodi neke kriptografske primitive; pametna kartica, itd.
- Vanjski izvor napajanja, generator pulsa i ostala električka oprema potrebna da uređaj radi ispravno.
- Sonda za mjerenje odljeva. Primjerice, otpornik male vrijednosti može se dodati u strujni krug napajanja te se tako može mjeriti potrošnja energije. Dodatno, EM zračenje može se mjeriti s jednostavnom ručno motanom zavojnicom.
- Uređaj za nadgledanje, npr. digitalni osciloskop s dovoljno velikom rezolucijom (tipično 1 GS/s, 8 bitova rezolucije, ...) spojen na računalo za statističku analizu tragova.

Mjerne sheme imaju veliki utjecaj na efektivnost napada SCA. Budući da je šum glavni problem napada SCA, kvaliteta mjerenja se uglavnom kvantificira pomoću količine šuma u tragovima. Idealno, mjerenje bi trebalo biti izvedeno tako da sve njime dobivene informacije budu relevantne za analizu. Ovo naravno u praksi nije moguće.

2.2. Klasifikacija napada

Napadi SCA uglavnom se klasificiraju u tri glavne skupine:

- Klasifikacija prema kontroli nad procesom računanja
- Klasifikacija prema načinu pristupanja modulu
- Klasifikacija prema metodama korištenim prilikom analize

2.2.1. Kontrola nad procesom računanja

Prema različitim načinima kontrole napadača nad procesom izvođenja računa napadi SCA dijele se na dvije glavne kategorije: ***pasivne*** i ***aktivne***. Ti napadi razlikuju se po utjecaju napadača na operativnost sustava. Kod pasivnih napada govorimo o napadu u kojemu napadač uspješno dohvati informacije o ciljanom sustavu, ali operativnost tog sustava ostaje nepromijenjena, kakva bi i bila da se napad nije dogodio. Primjer takvog napada bilo bi prisluškivanje. Aktivni su napadi, s druge strane, oni kod kojih se direktno utječe na rad sustava. Ciljani sustav može, ali ne mora moći detektirati da je došlo do napada, no vanjski promatrač može vidjeti promjenu u radu. Valja napomenuti da u ovakvom tipu napada nije zahtijevano da napadač dohvati informacije, već da samo ometa ciljani rad uređaja. Primjer ovakvog napada bila bi izmjena već poslane poruke.

2.2.2. Načini pristupa modulu

Prilikom analiziranja sigurnosti kripto procesora može biti korisno provesti sistematski presjek površine napada – skupa fizičkih, električkih i logičkih sučelja koji su izloženi potencijalnom napadaču. Vođeni ovom misli, Anderson et al. (2006) podjelili su napade na sljedeće četiri skupine.

Invazivne napade (engl. *Invasive attacks*) To su oni napadi koji zahtijevaju direktan pristup unutarnjim komponentama kripto procesora. Naprimjer, napadač napravi rupu u unutrašnjost mikrokontrolera te direktno mjeri podatke s fizičke sabirnice. Iako je sondiranje dijelova mikročipa uvijek bilo dostupno proizvođačima poluvodičkih uređaja, sredinom 90-ih počela je pojava polovne opreme za testiranje poluvodiča na tržištima što je omogućilo invazivne napade široj populaciji. Tipična stanica za sondiranje sastoji se od mikroskopa montiranog na podlozi s minimalno vibracija. Mikroskop bi tipično sadržavao i laser s kojim bi se probijale rupe kroz pakiranje mikroprocesora. Proizvođači su razvili razne metode otpora na invazivne napade koje prilikom pokušaja napada trajno onesposobe uređaj. Također, kako napredak tehnologije omogućuje

da veličina komponenata korištenih u proizvodnji bude sve manja invazivni napadi postaju nepraktičniji i teži.

Polu-invazivne napade (engl. *Semi-invasive attacks*) Takvi napadi uključuju pristup uređaju, ali bez potrebe za oštećenjem uređaja. Najraniji primjer ovakvog napada koristio je UV svijetlo kako bi se resetirao zaštitni bit na mikrokontroleru, kako bi se mogao pročitati sadržaj memorije. Još jedan od ranijih primjera bila je tehnologija testiranja poluvodiča prilikom koje je osvjetljavanjem nevodljivog CMOS tranzistora rezultiralo s mjerljivom strujom odljeva. Polu-invazivni napadi su postali praktični tek početkom 21-og stoljeća pojavom tehnika optičkog sondiranja. Ideja je da osvjetljavanjem tranzistora on postane provodljiv te da se tako uvede greška u njegovom radu. Takvi napadi mogu biti provedeni s jednostavnom opremom vrlo niske cijene. Korištenjem laserskih sondi, moguće je pasivno i čitanje memorijskog sadržaja ciljanog uređaja. To se postiže tako da se svaka memorijska ćelija osvijetli jedna po jedna, a rezultirana ionizacija prouzroči mjerljivu struju ako je sadržaj ćelije prazan.

Lokalno neinvazivne napade (engl. *Local non-invasive attacks*) Ovakav tip napada uključuje promatranje ili manipulaciju rada uređaja koji se nalazi u neposrednoj blizini. Primjer ovakvog napada je mjerenje struje potrebne za rad procesa s visokom preciznošću te određivanje korelacije s trenutnim računanjem koje se izvodi kako bi se došlo do podataka, kao što je opisano u odjeljku 2.1.1. Lokalno neinvazivni napadi spadaju u još generalniju skupinu napada, neinvazivne napade (engl. *non-invasive attacks*) zajedno s udaljenim napadima opisanim u sljedećem paragrafu. Bitna karakteristika neinvazivnih napada je da su potpuno neopazivi. Dodatno, ovakvi napadi uglavnom su jeftiniji od invazivnih napada, zbog čega predstavljaju veću prijetnju. Ovakav tip napada razmatrat će se u kasnijim poglavljima ovog rada.

Udaljene napade (engl. *Remote attacks*) Prilikom napada ove vrste, rade se opservacije ili manipulacije normalnih ulaza i izlaza uređaja, a sam napad je neovisan o udaljenosti napadača od uređaja. Primjeri ovakvog napada uključuju vremensku analizu, kriptoanalizu i napade na aplikacijsko programsko sučelje. U ovu skupinu spadaju i tradicionalni kriptografski napadi koji iskorištavaju mane u kriptografskim primitivama i sigurnosnim protokolima. Kao što je već spomenuto, ovakvi napadi također spadaju u neinvazivne napade.

2.2.3. Metode korištene prilikom analize

Prilikom podjele prema metodama korištenima prilikom analize podataka, SCA se dijeli na **jednostavnu** SCA (engl. *Simple Side-Channel Analysis*, dalje SSCA) i **diferencijalnu** SCA (engl. *Differential Side-Channel Analysis*, dalje DSCA).

SSCA karakterizira to da izlaz uglavnom ovisi o *trenutnoj* operaciji koja se izvodi. Kod SSCA, tipično se koristi samo jedan trag (engl. *trace*) iz kojeg se tajni ključ može direktno odrediti. Zbog ovakvog pristupa SSCA je vrlo osjetljiv na šum, informacije iz sporednih sredstva koje nisu u korelaciji s podacima. Dakle, bitno je da željeni signal, onaj povezan s traženim podacima bude izraženiji od šuma.

S druge strane, DSCA koristi korelaciju između procesiranih podataka i informacija sporednih sredstava. Kod DSCA generalno se koristi više tragova, a zatim nad njima provodi statistička analiza kako bi se odredile informacije o ključu. DSCA se najčešće sastoji od sakupljanja informacija, a zatim njihove analize. Kako bi se efikasno analizirala korelacija podataka sa stvarnim, mjerljivim, informacijama koristi se hipotetski model uređaja koji je pod napadom. Taj model se koristi kako bi se predvidjela sporedna sredstva uređaja; to može biti samo jedan izvor informacija ili se može koristiti za modeliranje više različitih izvora. Ako se trag koristi samo jednom govorimo o **napadu prvog reda** (engl. *first-order attack*), a ako se koristi dva ili više puta govorimo o **napadima drugog reda** (engl. *second-order attack*) i **napadima višeg reda** (engl. *higher-order attack*), respektivno (Zhou i Feng, 2005).

2.3. Postojeći napadi

2.3.1. Napadi analizom trajanja

Napadi analizom trajanja (engl. *timing attacks*) su oni napadi koji se baziraju na tome da se, zbog optimizacija, kriptografski algoritmi uglavnom izvode operacije u nekonstantnom vremenu. Varijacije u trajanju izvođenja, dakle, mogu sadržavati informacije o operacijama koje su u tijeku. Ovu ideju predstavio je Kocher (1996), a praktično su je izveli Dhem et al. (1998) koji su pokazali da je moguće osobnim računalom tadašnjice dekriptirati 512-bitni ključ u nekoliko sati s 300 000 mjerenja te 128-bitni ključ s 10 000 mjerenja, čime su pokazali ozbiljnost ovakvih napada. Brumley i Boneh (2005) su zatim pokazali da je, koristeći vremenske napade, moguće otkriti RSA privatne ključeve iz web servera baziranog na popularnom OpenSSL-u preko lokalne mreže. Pokazali su kako modificirana verzija Kocherovog napada može biti uspješno

izvedena u ovakvom scenariju, tako što su napravili $\sim$1/3 milijuna upita ($\sim$2 sata).

Jedna od jednostavnijih obrana je da parametri operacije ne budu ovisni o ulaznim podacima. Primjerice, tako da sa slučajno generiranim podacima transformiramo parametre prije operacije, a zatim poništimo obrnutom operacijom kasnije. Druga vrsta obrane je eliminacija grananja u računu kako bi trajanje bilo jednako za vrijednosti podataka. Iako ovakav način uspješno uklanja mogućnost ovakvog napada, on ujedno i donosi velika kašnjenja u vremenu izvođenja. Noviji način rješavanja ovog problema je dizajniranje samog sklopovlja da sve operacije traju jednako, neovisno o tome koji podaci se obrađuju.

2.3.2. Napadi uzorkovanjem kvarova

Pokazalo se da sklopovne greške i kvarovi nastali za vrijeme operacija kriptografskog sustava mogu dovesti do ozbiljnih sigurnosnih propusta. Ponašanje sustava za vrijeme stanja kvara može pružiti uvid u njegov rad i dati informacije potrebne za uspješno dohvaćanje prividno sigurnih informacija. Napadi uzorkovanjem kvarova (engl. *fault attacks*) predstavljaju praktične i efektivne napade protiv kriptografskog sklopovlja kao što su pametne kartice, a pomoću njih su gotovo svi kriptografski algoritmi probijeni. Napadi uzorkovanjem kvarova jako variraju ovisno o ciljanom modulu te mogućnostima napadača i tipu kvara koj može biti izazvan.

Postoje dvije velike skupine sporednih sredstava omogućenih kvarovima. Prvo sredstvo je slanje namjerno iskvarenih podataka s ciljem da modul ne može uspješno rukovati takvim podacima. Drugo sredstvo je mogućnost stavljanja modula u ekstremne uvjete te tako uvesti kvar u njega. Neke od mogućih izvora smetnji mogu biti nagla promjena napona, temperature, svijetla, brzine pulsa...

Napadi uzorkovanjem kvarova, predstavljeni prvi puta od Boneh et al. (1997), opisani su kao teoretski model napada na RSA potpise i identifikacijske protokole Fiat-Shamir i Schnorr.

Jedna od najjednostavnijih protumjera je, ako pretpostavimo da napadač ne može dva puta uvesti isti kvar, provesti neke operacije simetričnih algoritama dva puta, no to će naravno umanjiti performanse. Ili u slučaju javnih ključeva, dodatno provjeriti potpis s privatnim ključem neposredno prije slanja. Dodatno su predloženi drugi načini zaštite kao što su dodavanje kontrolne sume i slučajnosti u izvođenje.

2.3.3. Napadi analizom snage

Kao što je već opisano, potrošnja energije kriptografskih uređaja prilikom rada može dati dodatne informacije o operacijama koje se izvršavaju i o uključenim podacima. Napadi koji iskorištavaju ovu činjenicu spadaju u skupinu napada analizom snage (engl. *power analysis attacks*). Napadi ove vrste pokazali su se iznimno uspješnima za većinu tradicionalnih implementacija simetričnih i asimetričnih algoritama te je u vrijeme pisanja ovog rada velika većina istraživanja u polju SCA upravo temeljena na njima. Polje analize snage dijeli se na **jednostavnu** analizu snage (engl. *Simple Power Analysis*, dalje SPA), **diferencijalnu** analizu snage (engl. *Differential Power Analysis*, dalje DPA) i **korelacijsku** analizu snage (engl. *Correlation Power Analysis*, dalje CPA).

U napadima SPA, cilj je pomoću analiziranja mjernih vrijednosti električne energije kroz vrijeme, odrediti koji uzorci vrijednosti predstavljaju koju instrukciju. Ideja je tako odrediti koja operacija se izvodi u koje vrijeme te koji su parametri te operacije. Primjerice, serija operacija XOR na osciloskopu izgleda drugačije nego serija operacija množenja. No, kompleksni sustavi uglavnom izvode više operacija u isto vrijeme, zbog čega se stvara dodatni šum koji otežava napade SPA, što nas dovodi do drugog tipa napada.

U napadima DPA, analiza snage provodi se i za vrijeme normalnih kriptografskih operacija te se ti podaci koriste kako bi se "oduzeli" od onih dobivenih analizom snage za vrijeme izvođenja kriptografskih operacija. Na taj se način može umanjiti šum i poboljšati mogućnost uspješnog napada. Ovaj pristup, iako kompliciraniji, sličan je uklanjanju pozadinske buke iz zvuka. Napade DPA i SPA razvili su Kocher et al. (1999), gdje su uspješno pokazali praktičan napad na implementacijsko sklopovlje algoritma DES.

Treća vrsta napada su napadi CPA. Oni su statistički napadi koji koriste Pearsonov koeficijent korelacije. CPA je novija metoda te ima brojne prednosti naspram DPA, jedan od kojih je to da zahtijeva manji broj tragova potrebnih za uspješan napad. Ovu metodu prvi su opisali Brier et al. (2004).

Neke od predloženih metoda obrane na razini sklopovlja jesu korištenje unutarnjeg izvora napajanja, uvođenje slučajnosti u redoslijed izvođenja instrukcija, uvođenje slučajnosti u preimenovanja registara i korištenje dva kondenzatora, jedan koji se puni od vanjskog izvora i drugi koji pruža napajanje uređaju. S programske strane, jedna od predloženih mjera je uvođenje slučajnih instrukcija kako bi se promijenio uzorak potrošnje energije.

Primjer napada DPA na algoritam DES

Radi boljeg razumijevanja opisani su koraci izvođenja napada DPA na primjeru DES algoritma (Standaert, 2010):

1. **Odabir ciljanog algoritma i implementacije.** Napadač prvo odredi algoritam (npr. DES) i ciljanu implementaciju (npr. ASCI ili FPGA) s koje namjerava dohvatiti podatke.

2. **Odabir izvora odljeva i mjerne sheme.** Napadač zatim odredi vrstu odljeva koji namjerava iskoristit (npr. potrošnja snage ili EM zračenje) i način na koji će mjerenjem pretvoriti taj odljev korisne podatke.

3. **Odabir ciljanog signala.** Napadi SCA uglavnom zasebno napadaju razne dijelove sustava, tj. tajnog ključa. Dakle, napadač bira koji dio ključa želi odrediti. Ovo može, naprimjer, biti šest bitova ključa koji ulaze u prvu DES-ovu supstitucijsku kutiju (S-box) S0.

4. **Odabir ulaznih podataka.** Ovisno o mogućnostima napadača, on može ili birati koji podaci ulaze u uređaj u obliku poruke ili ako to nije moguće pretpostavlja se da može nadzirati podatke koji ulaze.

5. **Određivanje unutarnjih vrijednosti algoritma.** Za poznate ulazne poruke, napadač odredi vrijednosti ovisne o ključu unutar ciljanog uređaja koje se izračunavaju tijekom izvođenja algoritma. Iz računskih razloga, korisne su samo vrijednosti koje su ovisne o malom dijelu ključa. Primjerice, moguće je predvidjeti prva četiri bita nakon permutacije u prvom krugu DES-a, za svaku od 64 moguće vrijednosti koje ulaze u S0. Kao rezultat ovoga napadač je predvidio unutarnje vrijednosti implementacije za q poruka i za svaku razred mogućih ključeva s^*, koje sprema u vektore $\boldsymbol{v}_{s^*}^{(q)}$.

6. **Modeliranje odljeva.** Za isti skup razreda mogućih ključeva dobivenih u prošlom koraku, napadač modelira funkciju stvarnog odljeva ciljnog uređaja. Primjerice, ako pretpostavimo da potrošnja CMOS uređaja ovisi o promjeni stanja, Hammingova težina ili udaljenost može biti korištena kako bi se predvidio odljev.

7. **Mjerenje odljeva.** Mjerenjem stvarnog odljeva uređaja dobije se vektor odljeva $\boldsymbol{l_q} = [l_1, l_2, \ldots, l_q]$. On sadrži q tragova koji odgovaraju q različitih ulaza.

8. **Odabir relevantnih tragova.** S obzirom na to da tragovi mogu sadržavati veliki broj uzoraka, generalno se oni filtriraju vizualno ili koristeći neke statističke metode. Kao rezultat ove faze dohvaćen je smanjeni vektor $R(\boldsymbol{l_q})$

9. **Statistička usporedba.** Konačno, za svakog kandidata razreda ključeva, napadač radi statističku usporedbu predviđenog odljeva sa stvarnim mjerenim podacima. Ako je napad uspješan, očekivano je da model za odgovarajući ključ iz razreda daje najbolje rezultate usporedbe.

2.3.4. Napadi analizom EM zračenja

Još jedan od opisanih sporednih sredstava je EM zračenje. Dakle, uređaji zbog svoje operacije emitiraju EM polje, a kada se elektromagnetska zavojnica postavi u to polje, zbog Faradayevog zakona, u njoj će biti inducirana električna struja. Takva, mjerljiva, struja daje napadaču uvid u rad uređaja. Jedan od popularnijih napada ove vrste naziva se RFID skimming, a uključuje očitavanje RFID informacija s beskontaktne kartice prilikom korištenja. Uređaji potrebni za ovakav napad potpuno su legalni te se mogu kupiti u normalnim trgovinama električke opreme. Dodatno, većina modernih pametnih telefona ima NFC čitač koji se može koristiti u iste svrhe. Područje koje se bavi ovim napadima naziva se elektromagnetska analiza (engl. *Electromagnetic Analysis*, dalje EMA) te je po mnogim svojstvima slično analizi snage. Isto kao i kod analize snage, postoje dvije kategorije napada: **jednostavna** EMA (engl. *Simple Electromagnetic Analysis*, dalje SEMA) i **diferencijalna** EMA (engl. *Differential Electromagnetic Analysis*, dalje DEMA), a one se razlikuju na prijašnje opisane načine. Postojanje ovakvih sporednih kanala, u vojnim je krugovima, poznato već dulje vrijeme, vidljivo iz deklasificiranog dokumenta (McConnell, 1992) američke nacionalne sigurnosne agencije (engl. *National Security Agency, NSA*) u kojem se raspravlja napad TEMPEST, daje se uputstvo instalacije opreme te se istražuju razna kompromitirana sredstva.

Protumjere protiv EM napada spadaju u dvije skupine, prva od kojih je smanjivanje jačine signala. Ovakva mjera uglavnom je na razini sklopovlja te uključuje redizajniranje uređaja i/ili stavljanje oklopa kako bi se umanjila količina EM zračenja vidljivog izvan modula. Druga skupina su metode smanjivanja količine informacije sadržane u signalu. To se postiže uvođenjem slučajnosti te čestim osvježavanjem ključa prilikom rada.

2.3.5. Napadi koji koriste zvuk

Zvuk je jedan od najranijih sporednih kanala, ali nedavno mu je ponovno pridodana pažnja kada su Shamir i Tromer (2004) pokazali da postoji korelacija između zvuka proizvedenog od procesora i operacija koje se izvode. Ovakvi napadi uglavnom ne moraju brinuti o akustičnom šumu okoline, poput ventilatora od računala, jer su ciljani zvukovi uglavnom frekvencija viših od 10 kHz. Zbog razlika u sklopovlju, temperaturama i ostalim okolnim uvjetima, zvukovi različitih računalnih jedinica uglavnom imaju različite zvučne potpise. Dodatno, snimanjem pritisaka tipki može se odrediti unos s tipkovnice (uključujući tipkovnice bankomata) na temelju različitih zvukova koji svaka tipka proizvodi. Pomoću pravog mikrofona, zvuk ispisnih glava printeru može se upotrijebiti za rekonstrukciju ispisanog sadržaja. Uz to, nedavno je pokazano kako se samo snimanjem zvuka zavojnice (engl. *coin whine*) u radu LCD monitora može utvrditi što se na njemu prikazuje.

Neke od predloženih protumjera jesu korištenje opreme za prigušivanje zvuka, kao što su zvučno izolirane kutije dizajnirane specifično kako bi umanjile relevantne frekvencije. Suprotno tome, dovoljno jak širokopojasni izvor šuma može maskirati informativne signale, iako ergonomske zabrinutosti mogu ovakvu mjeru učiniti neprivlačnom.

2.3.6. Napadi koji koriste vidljivu svjetlost

Kuhn (2002) je demonstrirao da je moguće rekonstruirati sliku s CRT monitora ako taj monitor nije u vidokrugu napadača, već sve što je dostupno je difuzni odraz s, primjerice, obližnjeg zida. Pretpostavio je dodatno da se ista tehnika može iskoristiti i za druge izvore vidljive svjetlosti poput LED dioda. Kasnije su Loughry i Umphress (2002) pokazali kako postoji odnos između statusnih LED dioda računala i podataka koji se koriste u računu.

2.3.7. Napadi koji koriste poruke o pogrešci

U mnogim standardima, poput SSL/TLS, IPSEC i WTLS, poruke su prvo formatirane, a zatim kriptirane ulančanim blokovskim šiframa (CBC). Na strani primatelja se za vrijeme dekripcije mora provjeriti valjanost formata. Informacija se lako odljeva iz komunikacijskih protokola u odabranom napadu, budući da primatelj uglavnom šalje potvrdu ili poruku o pogrešci. Ovo može postati korisno sporedno sredstvo za kriptoanalizu, a napadi koji ga koriste nazivaju se napadi koji koriste poruke o pogrešci

(engl. *Error message attack*). U praksi ovakva sredstva se pojavljuju jer napadač ima mogućnost da uvede predvidljive modifikacije u originalnoj poruci kroz modifikaciju kriptirane poruke. Ovakvi napadi pokazali su se uspješnima i kod simetričnih i asimetričnih kriptografskih algoritama. Najistaknutiji i uvjerljivi primjer napada koji koriste poruke o pogrešci vjerojatno je Bleichenbacherov (1998) napad na sustav RSA. Zbog toga konsenzus je da je bitno uključiti snažnu provjeru integriteta u RSA enkripciju. Kod novijih inačica većine modernih algoritama, utjecaji ovakvog napada većinski su uklonjeni.

2.3.8. Napadi bazirani na priručnoj memoriji

Velika većina modernih procesora, uz primarnu i sekundarnu memoriju, implementira i treći tip memorije; priručnu memoriju (engl. *cache*). Ta memorija uglavnom je implementirana u samom procesoru te služi kao međuspremnik između procesora i radne memorije. Ona se koristi zbog činjenice da je brzina procesora puno veća od brzine radne memorije. Ako priručne memorije ne bi bilo procesor bi često morao čekati podatke iz radne memorije, a ta bi komunikacija postala usko grlo (engl. *bottleneck*) sustava. No uz korištenje priručne memorije podaci za koje se očekuje pristup spremaju se u priručnu memoriju i tako se znatno smanjuje vrijeme čekanja podataka i ubrzava rad sustava. Ako procesor zatraži podatak koji se nalazi u priručnoj memoriji, to nazivamo pogotkom (engl. *cache hit*), a ako to nije slučaj to nazivamo promašajem (engl. *cache miss*). Promašaji priručne memorije iznimno su bitan pojam u polju programske optimizacije, ali pokazalo se da mogu biti vrlo bitni i u polju SCA. Naime, ako se dogodi promašaj, generirat će se kašnjenje u izvođenju zbog navedenih razloga. Mjerenje trajanja tog kašnjenja napadaču omogućuje da sazna kada se promašaj dogodio te frekvenciju pojavljivanja istih. Upravo to je ideja napada baziranih na priručnoj memoriji (engl. *Cache-based attacks*). Originalna ideja ovakvog napada je predložena od strane Kelsey et al. (1998). Napadi "Spectre" i "Meltdown" spomenuti u uvodu, jedni su modernijih napada koji koriste slabosti nastale zbog priručne memorije.

Razne mjere obrane su predložene, poput uklanjanja priručne memorije, ali efikasno rješenje koje ne ovisi o izvedbi je još uvijek otvoreno pitanje.

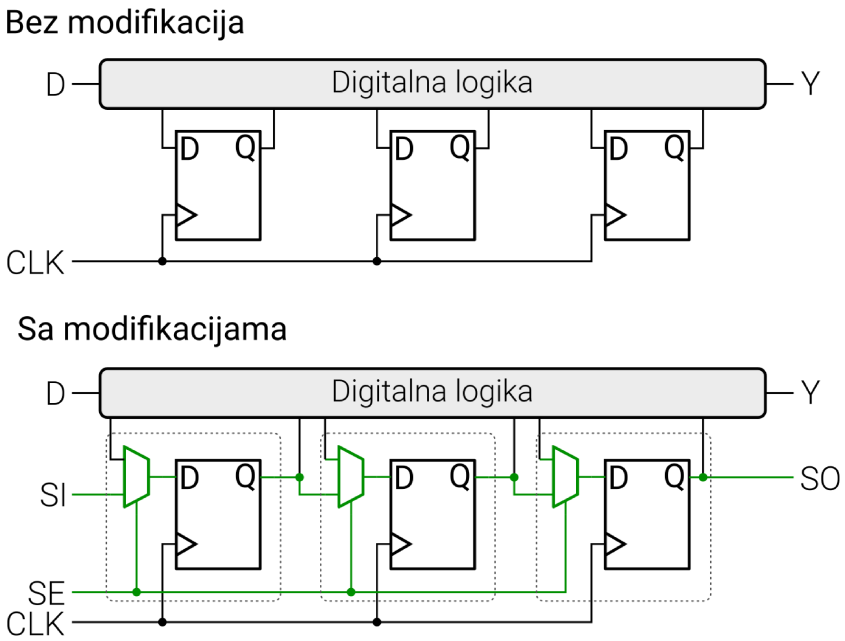
2.3.9. Napadi bazirani na frekvenciji

Ideju napada baziranih na frekvenciji predložio je relativno nedavno Tiu (2005). Metoda je učinkovita čak i kad tragovi nisu poravnati u eksperimentu, uvjet koji nije bio

ispunjen dotadašnjim DEMA istraživanjima. Glavne mete ovakvog napada su prijenosni uređaji, poput mobitela i digitalnih osobnih asistenata. Dodatno, za predloženu diferencijalnu frekvencijsku analizu (engl. *Differential Frequency Analysis*, dalje DFA) prvog reda, pokazano je da je uspješna usprkos dodavanju slučajnog kašnjenja. No, ova je metoda neuspješna kada se u protokol AES implementira protumjera razdijeljenom maskom (engl. *Split Mask*).

2.3.10. Napadi bazirani na skeniranju

Kako su logički uređaji postajali sve kompleksniji, zahtijevalo se puno više vremena i truda potrebnog za ručno stvaranje i validiranje testova. Također je bilo sve teže odrediti koji postotak funkcionalnosti test pokriva, a sami testovi trajali su predugo. Ovako opisana tehnika testiranja naziva se funkcijsko testiranje. No zbog nepraktičnosti, industrija je krenula s takozvanim "dizajniranje za testiranje" (engl. *design to test, DFT*) pristupom, gdje je sam dizajn modificiran kako bi bilo jednostavnije testirati sklopovlje. De facto norma ovakvog testiranja je test "skeniranjem" (engl. *scan test*). Bistabili sadržani u dizajnu modificirani su kako bi im se omogućilo da za vrijeme testiranja funkcioniraju kao stimulans i točke na kojima se mogu validirati ispravnosti, a da i dalje za vrijeme normalne operacije izvršavaju svoju standardnu ulogu.



Slika 2.4: Ne modificirani i modificiran dizajn

Budući da test skeniranjem mijenja bistabile koji su već uključeni u dizajn kako bi omogućio da djeluju i kao stanice za skeniranje, utjecaj testnog kruga je relativno

mali, obično se dodaje samo oko 1-5% na ukupan broj vrata. (Semiconductor Engineering, 2019) Očito je da su testovi skeniranja bitni za razvoj digitalnih uređaja, no isto tako Bo Yang et al. (2004) pokazali su da je i korisno sporedno sredstvo u tzv. napadima baziranim na skeniranju (engl. *scan-based attacks*), tako što su iskorišteni već postojeći skenirajući lanci za dobivanje tajnog ključa iz sklopovske implementacije algoritam DES algoritma. Naravno, s pažljivim dizajniranjem i odspajanjem lanaca nakon testiranja ovakve je napade moguće izbjeći.

2.4. Protumjere

Kako su se napadi razvijali, tako su i razvijale protumjere za njih. Postoji mnogo raznih strategija obrane od napada koji koriste sporedna sredstva, a neke generalne su:

- ukloniti korelaciju između izlaznih tragova (npr. pomoću dodavanja slučajnih vremenskih pomaka, umetanja lažnih instrukcija, uvođenja slučajnosti u redoslijed izvođenja, itd..);
- zamijeniti kritične instrukcije s onima čiji "potpis" nije jednostavno analizirati ili redizajnirati sklopovske komponente koje izvode aritmetičke operacije i/ili prebacivanje podataka iz memorije;
- napraviti algoritamske promjene kriptografskih primitiva kako bi napadi bili neučinkoviti na dobivenoj implementaciji (npr. maskiranje podataka i ključa sa slučajnom maskom generiranom pri svakom pokretanju)

Pokazano je da su algoritamske tehnike među svim tim vrstama protumjera najsvestranije, sveprožimajuće i mogu biti najefikasnije. Također, u mnogim su kontekstima one najjeftinije za provedbu. Programske mjere uključuju uvođenje lažnih instrukcija, uvođenje slučajnosti u redoslijed izvođenja, uravnoteživanje Hammingove težine unutarnjih podataka i dijeljenje bitova. Na razini sklopovlja, protumjere obično uključuju uvođenje slučajnosti u takt, uvođenje slučajnosti u potrošnju energije ili njenu kompenzaciju, uvođenje slučajnosti u izvršavanje instrukcijskog skupa i/ili korištenja registara. Ovakve mjere uglavnom su mnogo skuplje za provedbu, a njihov se učinak može umanjiti raznim tehnikama obrade signala. Programske mjere protiv napada SCA znatno ometaju izvedbu kriptografskih algoritama u pogledu memorije ili vremena izvođenja ili oboje. Jedan od glavnih izazova je postići sigurnu implementaciju sa što manje dodatnih troškova.

Odabir odgovarajuće razine protumjera može ovisiti o vrijednosti ciljanih podataka i moći protivnika (npr. njegovom znanju, resursima, itd.). Procjenjivanje razine otpora

treba obaviti barem iz sljedeća tri ugla: moć protivnika (uključujući njegovo znanje, resurse i vještine i dr.), moć napada (koja je usko povezana s najsuvremenijom tehnologijom) i djelotvornost protumjera. Do sada se pokazalo da kombinacija sklopovskih i algoritamskih mjera zaštite daje vrlo dobar omjer sigurnosti i troškova. (Zhou i Feng, 2005).

2.4.1. Uvođenje slučajnosti

U napadima SCA, napadač pokušava povezati uzorke inherentne sporednim svojstvima s operacijama i podacima koji se obrađuju. Kako bi se to otežalo, jedna metoda je uvesti slučajnost podatke koji istjecaju kroz razne sporedna sredstva, kao što su potrošnja energije, EM zračenje ili duljina trajanja izvršavanja. Ideja je tako, sa strane napadača, smanjiti korelaciju te tako i mogućnost dobivanja korisnog znanja o unutarnjem radu sustava. Slučajnost se može uvesti u razne dijelove sustava na mnogo načina.

U slučaju kriptosustava koji koriste eliptične krivulje, metoda slučajnih projektivnih koordinata je praktična protumjera protiv napada SCA u kojima napadač ne može predvidjeti pojavu određene vrijednosti jer su koordinate slučajno generirane. U ovakvom slučaju standardni DPA koristi korelacijsku funkciju koja može razlikovati je li određeno sredstvo povezano s opaženom operacijom. Kako bismo to izbjegli potrebno je uvesti slučajnost i parametre eliptičkih krivulja.

2.4.2. Zasljepljivanje

Zasljepljivanje je koncept u kriptografiji koji omogućuje klijentu da postoji poslužitelj koji računa matematičku funkciju $y = f(x)$, gdje klijent pruža ulaz x i dohvaća odgovarajući izlaz y, ali poslužitelj ne bi saznao ništa o x ni y. Ovaj koncept koristan je kada klijent ne može računati matematičku funkciju f sam (Bleumer, 2011).

Kao primjer, sagledajmo sljedeću situaciju: klijent želi dekriptirati podatak x, tj. želi odrediti $y \equiv x^d \pmod{n}$, uz ulaz x. Kako poslužitelj ne bi mogao znati ni x, a ni y moramo iskoristiti tehnike zasljepljivanja; klijent odabere slučajan broj r, a zatim odredi $s \equiv r^e \pmod{n}$. Tada je podatak koji šalje poslužitelju $X \equiv xs \pmod{n}$. Poslužitelj zatim računa $Y \equiv X^d \pmod{n}$, a izlaz Y zatim dohvaća klijent, koji ga pretvara u $y \equiv \frac{Y}{r} \pmod{n}$. Vidljivo je da vrijedi

$$y \equiv \frac{Y}{r} \equiv \frac{X^d}{r} \equiv \frac{(xs)^d}{r} \equiv \frac{x^d s^d}{r} \equiv \frac{x^d r}{r} \equiv x^d \pmod{n}, \tag{2.3}$$

što je upravo ono što smo htjeli izračunati, ali poslužitelj je znao samo vrijednost X i Y, bez poznavanja x i y.

Tehnike zasljepljivanja su najefektivnija mjera obrane napada analizom potrošnje energije i napada koji koriste trajanje.

2.4.3. Maskiranje

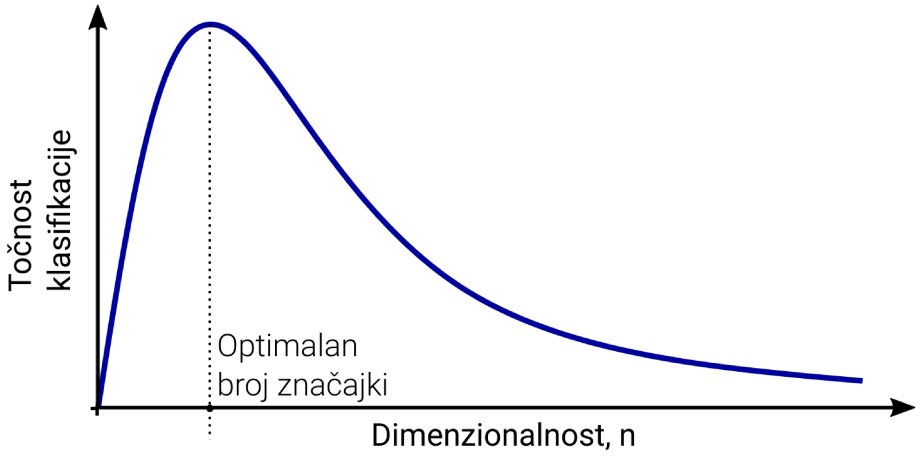
Maskiranje podataka je proces sakrivanja originalnih podataka pomoću modificiranog sadržaja. Ta tehnika je najrasprostranjenija protumjera protiv napada analizom potrošnje energije i napada koji koriste trajanje na programskoj razini. Ona je također jedna od najboljih programskih protumjera protiv napada SCA. Maskiranje podataka zasniva se na jednostavnoj ideji, da se poruka i ključ maskiraju sa slučajno generiranom maskom na početku izračuna, nakon čega se ostatak izvodi jednako kao i bez maske. Naravno, na kraju je potrebno da maska bude poznata kako bi se podaci mogli obnoviti. Taj proces naziva se korekcija maske (engl. *mask correction*).

3. Smanjenje dimenzionalnosti

Smanjenje dimenzionalnosti (engl. *dimensionality reduction*) je jednostavno rečeno proces smanjivanja dimenzija skupa značajki.

Kada pričamo o podacima iz stvarnog svijeta, skup podataka uglavnom je sastavljen od velikog broja značajki. Smanjenjem dimenzionalnosti pomažemo ukloniti nepotrebne značajke, smanjujemo računalne zahtjeve (vremenske i prostorne) i uklanjamo šum. Time omogućavamo bolju generalizaciju te smanjujemo mogućnost prenaučenosti (engl. *overfitting*) klasifikatora koji treba te podatke klasificirati u razrede.

Hughesov fenomen pokazuje da kako broj značajki raste, raste i učinak klasifikatora sve dok ne dosegnemo optimalni broj značajki. Dodavanje više značajki će dalje smanjivati učinak, vidljivo na slici 3.1. Ovaj fenomen prvi je objavio i detaljno objasnio Hughes (1968) te je po njemu dobio ime. Intuitivno objašnjene je da kako bi naučili dobar sustav za predviđanje on mora "vidjeti" puno mogućih kombinacija vrijednosti kako bi naučio uzorke u podacima. S više značajki postoji puno više kombinacija tako da je potrebno i puno više primjera. Hughes je također pokazao da ukoliko bi imali beskonačan izvor primjera rast dimenzija bi kontinuirano rezultirao i rastom točnosti klasifikatora, no to naravno nije moguće u stvarnom svijetu.



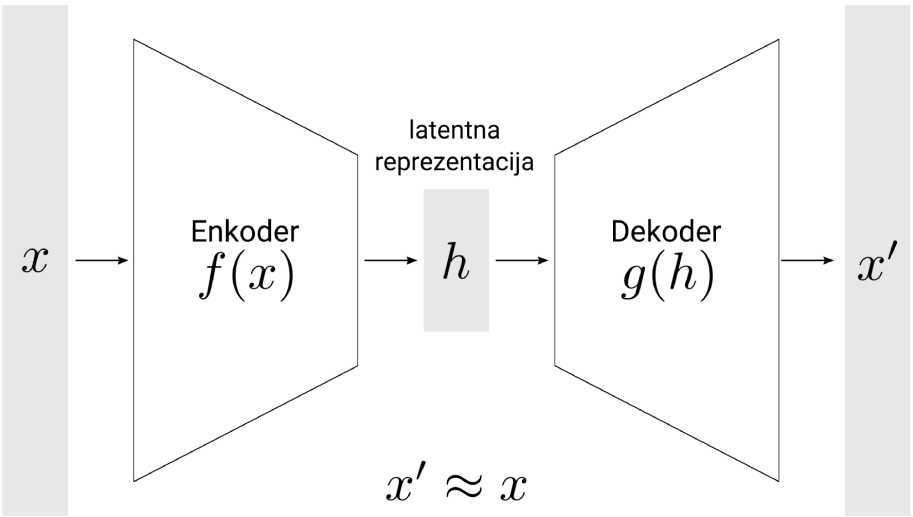
Slika 3.1: Odnos dimenzionalnosti i točnosti klasifikacije

Ovo je ujedno i oblik tzv. prokletstva dimenzionalnosti. Dakle, kako naša dimenzionalnost raste u jednom trenutku sposobnost generalizacije klasifikatora pada.

Smanjenje dimenzionalnosti može se napraviti na razne načine. Najosnovniji od njih je odabir značajki (engl. *feature selection*). Odabir se može izvoditi ili ručno ili algoritamski. Jedan od popularnijih primjera je postavljanje praga na varijancu (engl. *variance threshold*). Dakle, ako značajka ima jako malu varijancu kroz primjere sigurno je pretpostaviti da neće mnogo pridonijeti predviđanju.

Osim odabirom značajki smanjenje se može raditi transformacijom prostora, koja može biti linearna ili nelinearna. Ako se radi o linearnoj transformaciji, govorimo o linearnom smanjenju dimenzionalnosti, neke od čijih metoda su PCA, analiza faktora (engl. *Factor analysis*) i LDA. Jednako, ako je riječ o nelinearnoj transformaciji prostora tada pričamo o nelinearnom smanjenju dimenzionalnosti. Nelinearno smanjenje dimenzionalnosti često se naziva i učenje mnogostrukosti. Ove dvije metode bit će objašnjene u odjeljcima 3.1 i 3.2 respektivno.

Još jedna vrlo zanimljiva metoda smanjenja dimenzionalnosti koja daje izvanredne rezultate su autoenkoderi. Autoenkoderi se u svojoj suštini sastoje od dvije neuronske mreže: prva, koja se naziva enkoder, uzima originalni ulaz $x \in \mathbb{R}^n$ te ga smanjuje u takozvanu latentnu reprezentaciju $h \in \mathbb{R}^m$, gdje je $m < n$. Druga mreža, pod imenom dekoder, pretvara izlaz prvog u idealno kopiju originalnog ulaza $x' \approx x$. Ovaj proces vidljiv je na slici 3.2.



Slika 3.2: Autoenkoder

3.1. Linearno smanjenje dimenzionalnosti

Linearno smanjenje dimenzionalnosti (engl. *linear dimensionality reduction*) je najraširenija metoda smanjenja dimenzionalnosti. Moguće ju je vidjeti u poljima poput statističke analize, strojnog učenja i primijenjenim područjima matematike kroz stoljeća. Postala je neophodan alat za analiziranje visokodimenzionalnih podataka s mnogo šuma. Ovakve metode stvaraju niskodimenzionalno linearno mapiranje originalnih visokodimenzionalnih podataka, ali tako da zadrže željene značajke.

Definicija 1 (Linearna transformacija) *Neka V i W budu vektorski prostori na istom polju K. Tada za funkciju $f : V \mapsto W$ kažemo da je linearna transformacija (mapiranje) ako uz $\forall u, v \in V$ i $\forall c \in K$, vrijede sljedeća svojstva:*

(i) $f(u + v) = f(u) + f(v)$

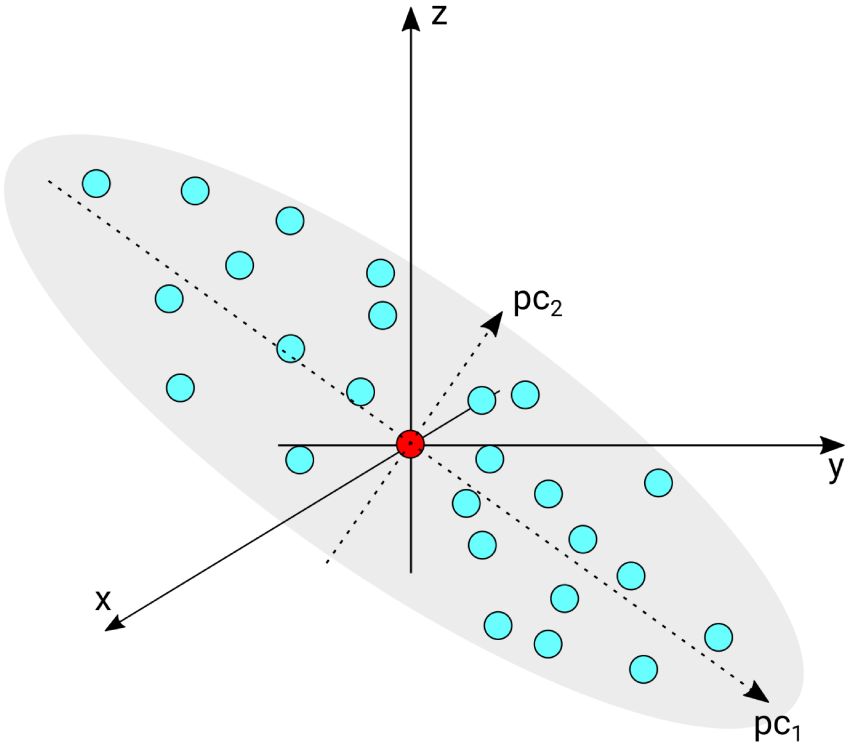
(ii) $f(cu) = cf(u)$

Dakle, može se reći da je linearna transformacija ona koja očuva operacije zbrajanja i množenja.

Postoje mnogo raznih metoda linearnog smanjenja dimenzionalnosti, samo neke od kojih su analiza glavnih komponenata (engl. *Principal Component Analysis, PCA*), analiza faktora (engl. *Factor Analysis, FA*), linearna diskriminantna analiza, (engl. *Linear Discriminant Analysis, LDA*), kanonička korelacijska analiza (engl. *Canonical Correlations Analysis, CCA*) i mnoge druge. Za potrebe ovog rada bit će obrađena PCA, ali pregled i uvid u druge metode linearne analize moguće je vidjeti u Cunningham i Ghahramani (2015).

3.1.1. Analiza glavnih komponenti – PCA

PCA je jedna od daleko najpopularnijih metoda smanjenja dimenzionalnosti koju je prvi put prezentirao Karl Pearson (1901) kao minimizaciju kvadratnog zbroja rezidualne pogreške između originalnih i projiciranih podataka. Moderne inačice PCA uglavnom rade ekvivalentnu maksimizaciju varijance. Statistički, PCA traži linije, ravnine i hiperravnine u n-dimenzionalnom prostoru tako da koordinate imaju najveću moguću varijancu podataka. Primjer je vidljiv na slici 3.3. Prva glavna komponenta (pc_1) je linija koja predstavlja smjer najveće varijance podataka. Za drugu komponentu (pc_2) vrijedi da je ortogonalna na prvu, da prolazi kroz centroid, a da pokazuje smjer drugog najvećeg smjera varijance. Jednako tako opisuju se i sve sljedeće komponente.



Slika 3.3: Primjer PCA

PCA se sastoji od nekoliko koraka. Prvi korak je normalizacija. Podaci se transformiraju kako bi imali srednju vrijednost μ jednaku 0 i standardnu devijaciju σ jednaku 1 prema jednadžbi 3.1 za svaki x.

$$z = \frac{x - \mu}{\sigma} \tag{3.1}$$

Time efektivno centroid naših podataka pomičemo u ishodište koordinatnog sustava. Na slici 3.3 ovaj korak je već izveden te je centroid označen crvenom bojom.

Sljedeći korak je odrediti glavne komponente. Kako bismo to odredili potrebno je prvo izračunati matricu kovarijance koristeći formulu 3.2.

$$cov(X, Y) = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \bar{x})(Y_i - \bar{y}) \tag{3.2}$$

Cilj ovog koraka je razumjeti kako se varijable skupa ulaznih podataka razlikuju od srednje vrijednosti u odnosu jedna na drugu ili drugim riječima kako bi se utvrdilo postoji li međusobna veza. Valja naglasiti da zbog činjenice da je $cov(x, x) = var(x)$, na glavnoj dijagonali kovarijancijske matrice nalaze se varijance originalnih značajki, a zbog toga što je operacija računanja kovarijance komutativna tj. vrijedi $cov(x, y) = cov(y, x)$, matrica je simetrična u odnosu na glavnu dijagonalu.

Nakon što znamo matricu kovarijance, potrebno je izračunati njene vlastite vektore (engl. *eigenvectors*) i vlastite vrijednosti (engl. *eigenvalues*) kako bismo odredili glavne komponente.

Definicija 2 *Za danu kvadratnu matricu $A \in \mathbb{R}^{n\times n}$, ponekad je moguće pronaći ne-nul vektor $v \in \mathbb{R}^n$ i odgovarajući skalar λ takav da vrijedi*

$$Av = \lambda v. \tag{3.3}$$

Tada za svaki ne-nul vektor v koji zadovoljava 3.3 zovemo vlastiti vektor od A, a odgovarajući skalar λ vlastitu vrijednost.

Očito je da jednadžba 3.3 može biti zapisana i kao

$$(A - \lambda I)v = 0. \tag{3.4}$$

Glavne komponente su nove varijable koje se grade kao linearne kombinacije početnih varijabli. Te se kombinacije izvode na takav način da su nove varijable (tj. glavne komponente) nekorelirane, a većina informacija unutar početnih varijabli se komprimira u prve komponente. Dakle, ideja je da n-dimenzionalni podaci rezultiraju s n glavnih komponenti, ali se maksimalno moguće informacija "stavlja" u prvu komponentu, zatim maksimalno preostalih informacija u drugu i tako dalje. Tako može se uzeti nekoliko prvih dimenzija, a ne izgubiti mnogo informacija. To je moguće jednostavno odrediti koristeći izračunate vlastite vektore i pripadne vlastite vrijednosti, tako da se "uzmu" vlastiti vektori koji imaju najveću pripadnu vlastitu vrijednost i od se njih napravi takozvani vektor značajki (engl. *feature vector*). Dakle, vektor značajki je jednostavno rečeno matrica koja kao stupce sadrži, poredane po pripadnoj vlastitoj vrijednosti, vlastite vektore koji su odlučeni biti zadržani. Drugim riječima, ako je cilj da konačna dimenzija prostora bude m tada će ta matrica imati m stupaca.

Osim standardnog PCA postoje i druge inačice kao što su iterativni PCA i nelinearni PCA koji koristi razne jezgrene (engl. *kernel*) funkcije.

3.2. Nelinearno smanjenje dimenzionalnosti

Nelinearno smanjenje dimenzionalnosti u literaturi često se naziva i učenje mnogostrukosti (engl. *manifold learning*). Za razumijevanje potrebno je prvo objasniti mnogostrukost.

Mnogostrukost U matematici, n-dimenzionalna topološka mnogostrukost ili samo mnogostrukost je apstraktni topološki prostor koji je lokalno euklidski, tj. za koji vrijedi da oko svake točke postoji susjedstvo koje je topološki jednako kao otvorena jedinična kugla u n-dimenzionalnom prostoru $\mathbb{R}^n$. Za ilustraciju ove ideje, dovoljno je sagledati vjerovanja da je Zemlja ravna ploča u kontrastu modernih dokaza da je okrugla. Ta nedosljednost uglavnom proizlazi iz činjenice da na malim mjerilima kojima ljudi vide Zemlju ona doista izgleda ravno. Generalno, svi objekti koji su približno "ravni" na malim mjerama su mnogostrukosti (Rowland). Za potrebe ovog rada, iako ne potpuno ispravno, dovoljno je gledati na mnogostrukost kao na generalizaciju jednostavne n-dimenzionalne površine.

Hipoteza o mnogostrukosti jedna je od najvažnijih hipoteza u cijelom strojnom učenju. Ona kaže da podaci iz stvarnog života, poput slika, leže na niskodimenzionalnoj mnogostrukosti ugrađenoj u visokodimenzionalni prostor. Intuicija kaže da bi ona trebala vrijediti jer slični podaci će u prostoru biti relativno blizu. Za primjer može se sagledati kamera koja se može rotirati oko dvije osi, tj. ima dva stupnja slobode te tako iz više položaja slika statični objekt. Uz pretpostavku da kamera uzima slike rezolucije 640x480, tada će svaka dobivena slika imati 307200 značajki, točnije ležat će u 307200-dimenzionalnom prostoru. No, očito je da te slike ovise o položaju naše kamere koji je moguće opisati samo s dvije značajke (rotacijom oko svake od osi) te imamo razloga vjerovati da naše slike leže na dvodimenzionalnoj mnogostrukosti koja je ugrađena u taj 307200-dimenzionalni prostor.

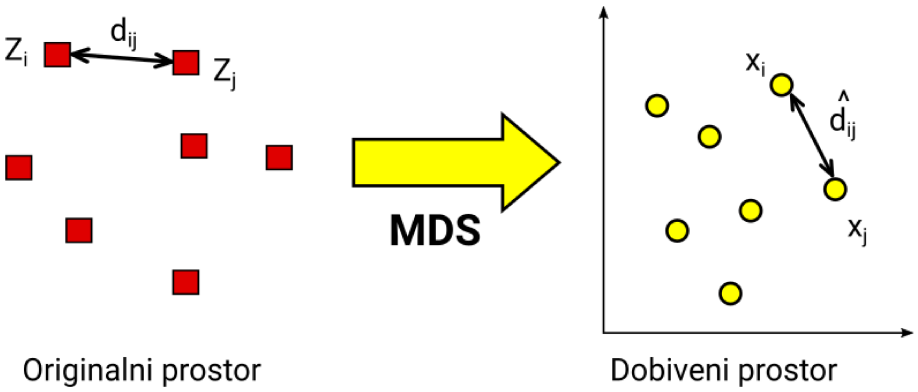
Učenje mnogostrukosti leži na hipotezi o mnogostrukosti. To je skupni naziv za metode pronalaženja niskodimenzionalne mnogostrukosti ugrađene u visokodimenzionalni prostor za koji hipoteza o mnogostrukosti tvrdi da postoji.

3.3. Algoritmi za učenje mnogostrukosti

Postoji mnogo algoritama za učenje mnogostrukosti, no ovdje ćemo navesti samo one implementirane u poznatoj knjižnici jezika python – scikit-learn. Dodatno, razmatrat će se i relativno novija metoda UMAP implementirana u pythonovoj knjižnici umap-learn.

3.3.1. Multidimenzionalno skaliranje – MDS

Multidimenzionalno skaliranje (engl. *Multidimensional scaling*, skrać. MDS) grupni je naziv za skup metoda koje predstavljaju mjere sličnosti između parova objekata kao udaljenost između točaka u niskodimenzionalnom prostoru. Podaci, naprimjer, mogu biti korelacija između objekata, tada je MDS reprezentacija ravnina koja prikazuje točke koje su bliže jedno drugoj što je njihova korelacija veća. U općem slučaju MDS može se koristiti za analiziranje sličnosti i različitosti u podacima, tako što sličnost ili različitost modelira kao udaljenost u geometrijskom prostoru (Borg i Groenen, 2005).



Slika 3.4: Prikaz preslikavanja MDS-a

Definicija 3 *Neka je matrica S matrica sličnosti, a X koordinate od n ulaznih točaka. Također, neka su razlike $\hat{d}_{ij}$ transformacija od S odabrana na neki optimalan način. Tada definiramo mjeru $Stress$ kao:*

$$Stress_r(X) = \sum_{i<j} (d_{ij}(x) - \hat{d}_{ij}(X))^2 \tag{3.5}$$

Ova mjera je ujedno i željena funkcija gubitka koja se minimizira. $Stress$ je mjera stupnja do koje udaljenost između uzoraka u smanjenom prostoru odgovara stvarnoj multivarijatnoj udaljenosti uzorka. Niže vrijednosti mjere $Stress$ ukazuju na veću

sukladnost i zato su poželjne. Visoke vrijednosti ukazuju da nije postojao niskodimenzionalni raspored točaka koji bi održavao njihove sličnosti. Generalno pravilo je da bi vrijednost ove mjere idealno trebala biti manja od 0.2 il čak 0.1. Ovisno o transformacijama koje se koriste kako bi se izračunali $\hat{d}_{ij}$, algoritmi MDS-a dijele se u dvije kategorije; metrički i nemetrički MDS.

Metrički MDS

Metrički MDS (engl. *Metric MDS*) je onaj kod kojeg ulazna sličnost matrice proizlazi iz metrike udaljenosti. Razmaci između dvije točke zatim su postavljeni da budu što je moguće bliži vrijednostima originalnih sličnosti podataka. Dakle, možemo reći da vrijedi:

$$\hat{d}_{ij} = f(S_{ij}). \tag{3.6}$$

Najčešće za tu funkciju vrijedi da je transformacija skalarom b, tj. vrijedi:

$$\hat{d}_{ij} = bS_{ij}. \tag{3.7}$$

U najosnovnijoj inačici metričkog MDS-a za skalar b vrijedi da je $b = 1$ i tada je očito da vrijedi:

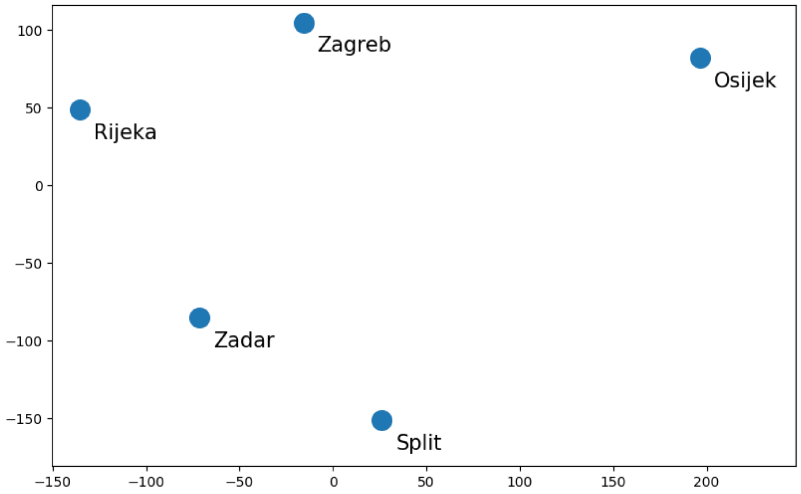
$$\hat{d}_{ij} = S_{ij}. \tag{3.8}$$

Takav MDS naziva se *apsolutni MDS*.

Na slici 3.5 može se vidjeti primjer metričkog mapiranja udaljenosti najvećih hrvatskih gradova specificiranih u tablici 3.1. Ovakvo preslikavanje samo je jedno od beskonačno mnogo različitih. Rezultantni prostor može biti, primjerice, zrcaljen, rotiran, a točke mogu biti i drukčije raspoređene. Jedino bitno svojstvo koje se čuva je međusobna udaljenost točaka.

Grad	Zagreb	Split	Rijeka	Osijek	Zadar
Zagreb	0	259	132	213	198
Split	259	0	257	289	118
Rijeka	132	257	0	333	148
Osijek	213	289	333	0	316
Zadar	198	118	148	316	0

Tablica 3.1: Udaljenost između gradova u Hrvatskoj



Slika 3.5: Metričko MDS preslikavanje gradova iz tablice 3.1

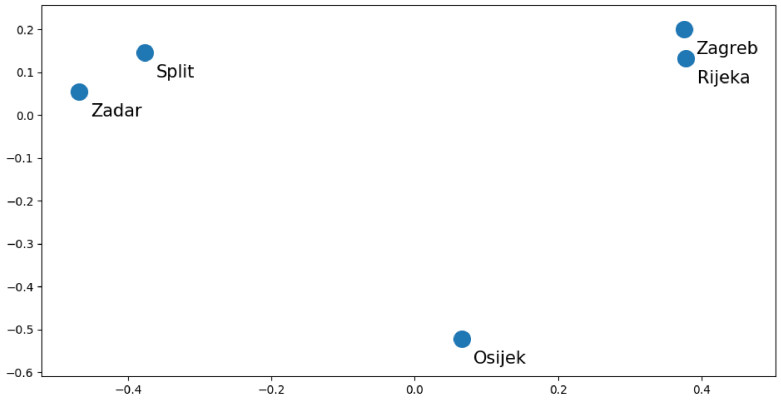
Nemetrički MDS

Glavna razlika nemetričkog MDS-a (engl. *Nonmetric MDS*) u odnosu na metrički je u tome što nemetrički pokušava sačuvati redoslijed udaljenosti originalnih podataka. Iz tog razloga traži monotonu vezu između udaljenosti u ugrađenom prostoru i sličnosti/različitosti.

Dakle, budući da se nemetrički MDS fokusira na redoslijeda u podacima, za njega možemo definirati sljedeću relaciju:

$$S_{ij} < S_{kl} \iff \hat{d}_{ij} < \hat{d}_{kl} \tag{3.9}$$

Na slici 3.6 vidljiv je rezultat istog primjera udaljenosti hrvatskih gradova iz tablice 3.1, ali izveden pomoću nemetričkog MDS-a.

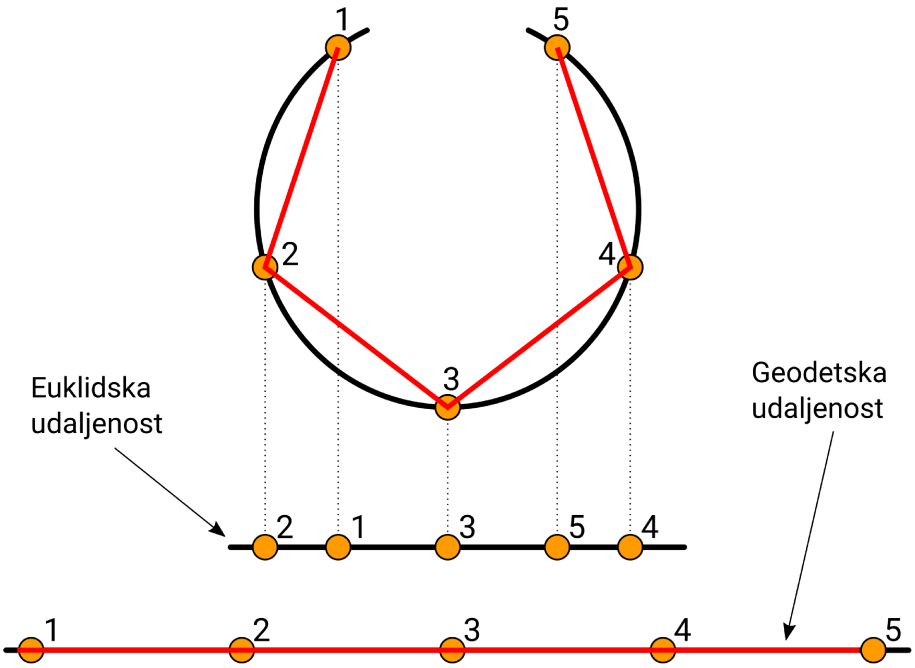


Slika 3.6: Nemetričko MDS preslikavanje gradova iz tablice 3.1

Vidljivo je kako udaljenosti nisu zadržane, ali je redoslijed gradova zadržan.

3.3.2. Izometrično preslikavanje – Isomap

Izometrično preslikavanje (engl. Isometric mapping) ili skraćeno Isomap je vrlo široko korištena metoda učenja mnogostrukosti, također jedna od najranijih. Ona je na neki način proširenje MDS-a. MDS radi preslikavanje koristeći euklidske udaljenosti parova primjera, dok Isomap koristi geodetske udaljenosti inducirane pomoću grafikona susjedstva ugrađenog u klasično skaliranje. Isomap definira geodetsku udaljenost kao zbroj težine bridova duž najkraćeg puta između dva čvora.



Slika 3.7: Razlika između euklidske i geodetske udaljenosti

Algoritam Isomap (u najkraćim crtama):

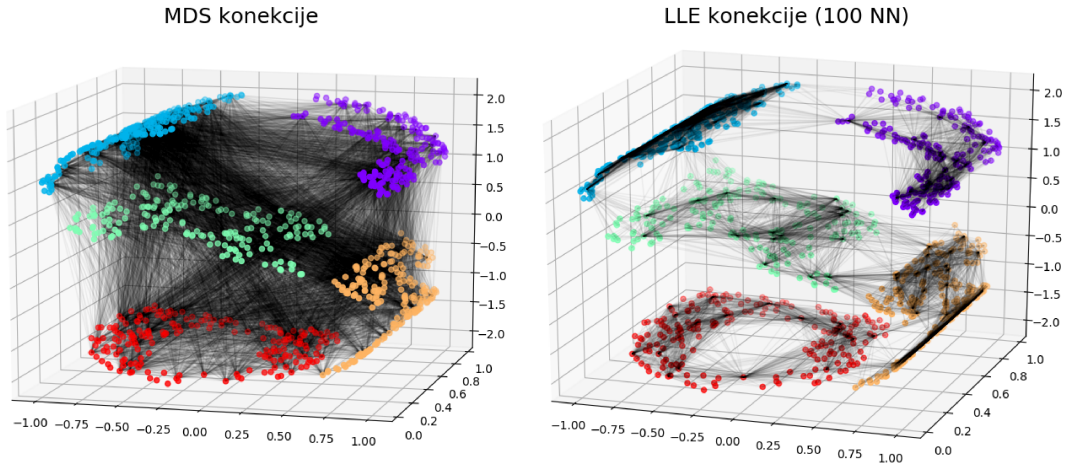
- odrede se susjedne točake za svaku točku (npr. k-NN, ball tree...)
- izgradi se susjedstvo (točke su povezane ako su susjedi, inače ne)
- računaju se najkraće udaljenosti između svih čvorova (npr: Dijkstra)

  $\rightarrow$ geodetske udaljenosti
- računa se niskodimenzionalna mnogostrukost (npr. MDS)

Zbog potrebe za računanje najbližih susjeda u svim točkama, a zatim najkraćih udaljenosti u grafu susjednosti, Isomap je računski vrlo zahtjevan za velike skupove podataka.

3.3.3. Lokalno linearno ugrađivanje – LLE

Lokalno linearno ugrađivanje (engl. *Locally Linear Embedding*, skrać LLE) je algoritam za učenje mnogostrukosti koji se temelji na jednostavnim geometrijskim intuicijama; da je mnogostrukost približno lokalno linearna na djelu oko točke. On za razliku od npr. MDS-a, koji gleda globalnu strukturu podataka, gleda samo lokalnu strukturu najbližih nekoliko susjeda zbog čega se bolje skalira na veće skupove podataka (slika 3.8).



Slika 3.8: Usporedba provjerenih udaljenosti algoritmima MDS i LLE

Pretpostavimo da se podaci sastoje od N vektora realnih podataka $\vec{X}_i$, svaki dimenzionalnosti D uzorkovanih s neke konveksne glatke temeljne mnogostrukosti. Pod uvjetom da postoji dovoljno podataka, za očekivati je da svaka točka podataka i njezini susjedi leže na ili vrlo blizu malog lokalno linearnog dijela temeljne mnogostrukosti. Tada se geometrija tog malog djela prostora može karakterizirati linearnom kombinacijom, tj. linearnim koeficijentima koji rekonstruiraju svaku točku iz svojih susjeda.

U najjednostavnijoj inačici algoritma LLE, prvi korak je odrediti lokalno susjedstvo točaka, što se radi pomoću algoritma k-najbližih susjeda. Pogrešku rekonstrukcije tada možemo mjeriti pomoću funkcije gubitka koju jednostavno definiramo kao:

$$\epsilon(W) = \sum_i |\vec{X}_i - \sum_{j \in N(i)} W_{ij} \vec{X}_j{}^2 \tag{3.10}$$

gdje je $N(i)$ susjedstvo od X_i.
Minimizacijom gubitka ϵ zatim se mogu odrediti težine W.

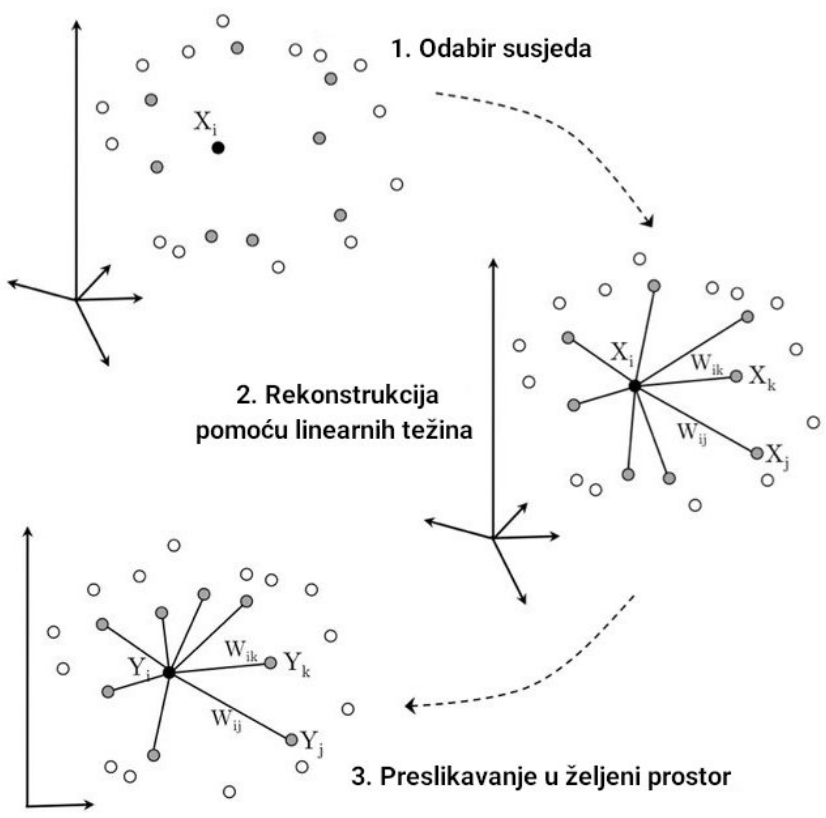
Konačno, koristeći dobivene težine, određuju se niskodimenzionalni vektori $\vec{Y}_i$ koji predstavljaju preslikanu vrijednost originalnih podataka opisanih vektorima $\vec{X}_i$. To se

radi tako da se minimizira gubitak koji definiramo sljedećom jednadžbom:

$$\Phi(Y) = \sum_i |\vec{Y_i} - \sum_j W_{ij}\vec{Y_j}^2 \tag{3.11}$$

Algoritam LLE (u najkraćim crtama):

- Odrediti susjede svake točke vektora $\vec{X_i}$
- Izračunati težine W_{ij} koje najbolje rekonstruiraju svaku točku $\vec{X_i}$ iz svojih susjeda minimizirajući funkciju gubitka ϵ definiranu jednadžbom 3.10
- Izračunati vektore $\vec{Y_i}$ koji su najbolja rekonstrukcija pomoću težina W_{ij} minimizirajući funkciju gubitka Φ definiranu jednadžbom 3.11



Slika 3.9: Ilustracija algoritma LLE

Izvor: Saul i Roweis (2003)

Odabir broja susjeda k je vrlo težak jer prilikom malih vrijednosti dolazi do nepovezanosti u grafu, a prilikom velikih vrijednosti dolazi do takozvanih rubova kratkog spoja (Balasubramanian et al., 2002). Iz tog razloga se pokazalo kako LLE nije uvijek topološki stabilan, tj. male promjene u povezanosti (topologiji) rezultiraju velikim promjenama u rezultatu.

3.3.4. Modificirano lokalno linearno ugrađivanje – MLLE

Modificirano lokalno linearno ugrađivanje (engl. *Modified Locally Linear Embedding*, skrać. MLLE) je poboljšanje algoritma LLE koji su predložili Zhang i Wang (2007) te je pokušaj rješavanja problema regularizacije metode LLE. MLLE to radi tako da za svaku točku koristi linearnu kombinaciju *više* vektora $\vec{W}_i$ za rekonstrukciju točke $\vec{X}_i$ u kontrastu sa samo jednim metode LLE. Ovom jednostavnom metodom uspješno se postiže veća stabilnost ugrađivanja.

3.3.5. Hessijsko svojstveno preslikavanje – HLLE

Hessijsko svojstveno preslikavanje (engl. *Hessian Eigenmapping*) također poznato kao LLE baziran na Hessijanu (skrać. HLLE) je još jedna metoda rješavanja regularizacijskog problema od algoritma LLE, uz dosadašnju pretpostavku konveksnosti mnogostrukosti. Ovaj algoritam se bazira na kvadratnoj formi $\mathcal{H}(f) = \int_M H_f(m)\|_F^2 dm$ definiranoj na funkciji $f : M \mapsto \mathbb{R}$. Ovdje H_f označava Hessijana od f. Za definiranje Hessijana, koriste se ortogonalne koordinate na tangentnom prostoru od M (Donoho i Grimes, 2003). Taj Hessijan se koristi u svakom susjedstvu za rekonstrukciju lokalne linearne strukture:

$$\mathcal{H}_{i,j} = \sum_l \sum_r ((H^l)_{r,i}(H^l)_{r,j}) \tag{3.12}$$

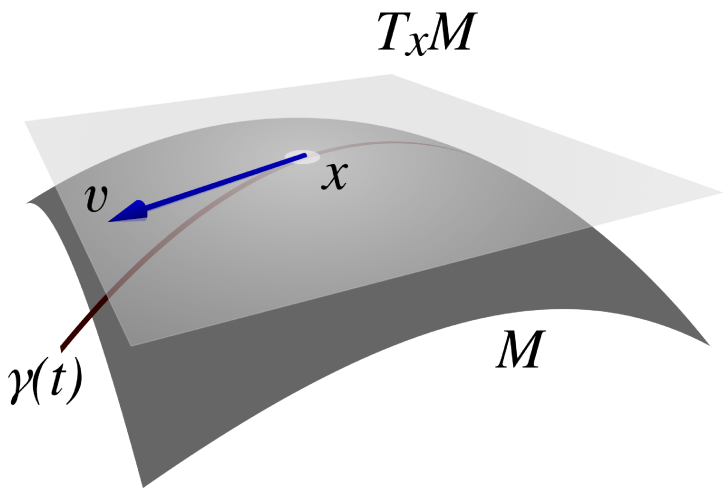
3.3.6. Spektralno ugrađivanje – SE

Spektralno ugrađivanje (engl. *Spectral embedding*) je tehnika ugrađivanja mnogostrukosti u niskodimenzionalni prostor koja koristi spektar (vlastite vrijednosti) matrice sličnosti. Ova metoda dolazi iz polja multivarijatne statistike i grupiranja podataka te se često koristi za određivanje grupa u podacima zbog čega se još i naziva spektralno grupiranje (engl. *spectral clustering*). Algoritam se sastoji od tri bitna koraka (Belkin i Niyogi, 2003):

- Izrađivanje matrice sličnosti pomoću težinskog grafa susjedstva (npr. k-NN) $\rightarrow A$
- Određivanje Laplacove matrice $L = D - A$, gdje je D matrica stupnjeva grafa susjedstva
- Dekompozicija svojstvenih vrijednosti (engl. *Eigenvalue decomposition*)

3.3.7. Lokalno tangentno poravnanje prostora – LTSA

Lokalno tangentno poravnanje prostora (engl. *Local Tangent Space Alignment*, skrać. LTSA) je metoda za pronalaženje ugrađene mnogostrukosti koja je vrlo slična metodi LLE (i izvedenim metodama kao HLLE i MLLE) te je bazirana na istoj pretpostavci, lokalne linearnosti. No, ova metoda, kao što je vidljivo iz imena, oslanja se na koncept tangentnog prostora T_xM mnogostrukosti M u točki x. Ukoliko bi se neki objekt kretao po mnogostrukosti u točki x bi imao neki vektor brzine koji je tangentan prostoru, drugim riječima to je tangentni vektor. Skupina takvih vektora iz mogućih krivulja koje prolaze kroz točku x definiraju tangentni prostor u njoj. Za 2D mnogostrukost ugrađenu u 3D prostor tangentni prostor bi bio ravnina (slika 3.10).



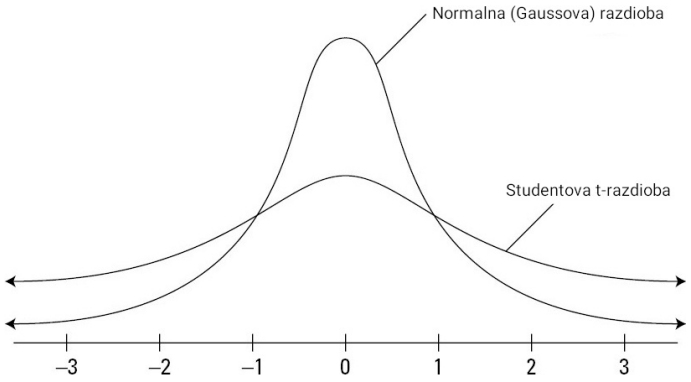
Slika 3.10: Tangentni prostor T_xM i tangentni vektor $v \in T_xM$, na krivulji koja prolazi $x \in M$.

Izvor: Wikimedia Commons (2008)

Ideja LTSA je da postoji linearno preslikavanje iz visokodimenzionalnog prostora u kojem leže originalni podaci u njegov lokalni tangentni prostor i da za odgovarajući niskodimenzionalni prostor postoji preslikavanje u isti taj lokalni tangentni prostor. Za LTSA i HLLE pokazano je da su matematički vrlo slični uz sitne razlike, poput toga da LTSA uključuje trenutnu točku u susjedstvo, dok HLLE ne te da iako koriste različite metode za rješavanje jednadžbe poravnanja, rješenje je jednako ako se uzme da HLLA koristi isti način izgradnje susjedstva kao LTSA (Zhang et al., 2018)

3.3.8. Ugradnja pomoću t-distribuiranog stohastičkog susjeda – t-SNE

Ugradnja pomoću t-distribuiranog stohastičkog susjeda (engl. *t-distributed Stochastic Neighbor Embedding*, skrać. t-SNE) relativno je novija metoda koja traži niždimenzionalnu strukturu tako da svojstva grupiranja ostanu sačuvana, gradeći neizrazitu (engl. *fuzzy*) topološku strukturu. Ovu metodu objavili su Maaten i Hinton (2008) kao metodu za vizualizaciju podataka visokodimenzionalnih podataka tako da se svakoj točki da lokacija niskodimenzionalnom skupu. To radi tako da pretvara srodstvo točaka u vjerojatnosti. Srodstva u izvornom prostoru predstavljena su Gaussovim zajedničkim vjerojatnostima, dok u ugrađenom prostoru su predstavljeni t-razdiobama. t-SNE metoda je varijacija ugrađivanja stohastičkog susjeda (engl. *Stochastic Neighbor Embedding*) koja je puno jednostavnija za optimizaciju te pruža vizualno puno bolje rezultate tako što smanjuje tendenciju podataka da se gomilaju u centru mape.



Slika 3.11: Prikaz normalne (Gaussove) razdiobe i Studentove t-razdiobe

t-SNE koristi Studentovu t-razdiobu za dobiveni prostor, a ne normalnu kao za originalni prostor, što je, zato jer normalna razdioba ima vrlo male vjerojatnosti pri rubovima (vidljivo na slici 3.11) te bi se podaci zgušnjavali prema sredini grupe. Studentova t-razdioba nema taj problem te to omogućuje da t-SNE bude posebno osjetljiv na lokalnu strukturu.

Kako bismo dobili distribuciju u ugrađenom prostoru Kullback-Leiblerova divergencija zajedničkih vjerojatnosti u izvornom prostoru i ugrađenom prostoru se minimizirana gradijentnim spustom. Treba primijetiti da KL divergencija nije konveksna, stoga je ponekad korisno program pokrenuti više puta koristeći drugo sjeme za ostvarenje slučajnosti. t-SNE je računalno vrlo zahtjevan te na uzorku s puno primjera potrebno je i nekoliko sati da završi te se iz tog razloga i svoje svojstvenosti da sačuva grupiranje uglavnom koristi za vizualizaciju podataka.

3.3.9. Uniformna aproksimacija i projekcija mnogostrukosti – UMAP

Posljednja velika metoda smanjenja dimenzionalnosti, naziva se uniformna aproksimacija i projekcija mnogostrukosti (engl. *Uniform Manifold Approximation and Projection*, skrać. UMAP), a objavili su je McInnes i Healy (2018). Ona pruža iznimne rezultate vizualno slične onima od t-SNE metode ali, zbog svoje snažne matematičke pozadine, ima mogućnost generalizacije na neviđenim podacima. Također, metoda UMAP puno bolje radi s većim skupovima podataka te je na, primjerice, skupu podataka "GoogleNews" 19 puta brža od t-SNE metode čije izvođenje traje 4.5 sati dok UMAP isti posao odradi u 14 minuta. Doduše, ova metoda pretpostavlja dva svojstva; da su podaci uniformno distribuirani i da je mnogostrukost lokalno povezana. Drugo svojstvo ne znači da je potrebno da cijela mnogostrukost bude povezana, već samo da ne postoji npr. točka koja je sama udaljena od svih. Ova pretpostavka je razumna za podatke iz stvarnog svijeta. Što se svojstva uniformnosti tiče, kako bi ga se osiguralo korisnik mora biti svjestan ovoga te po potrebi napraviti predobradu početnog skupa podataka. UMAP gradi graf susjedstva pomoću stabla slučajnih projekcija (engl. *Random projection trees*, skrać. RP-trees) i spust najbližih susjeda (engl. *Nearest Neighbor Descent*, skrać. NN-descent). Ovi vrlo efikasni algoritmi omogućuju brz pronalazak najbližih susjeda i izgradnju grafa.

Umjesto Studentove t-distribucije za modeliranje udaljenosti u niskodimenzionalnom prostoru, UMAP koristi sličnu obitelj krivulja definiranu sljedećom jednadžbom:

$$q_{ij} = (1 + a(y_i - y_j)^{2b})^{-1} \tag{3.13}$$

Parametre a i b pronalazi pomoću nelinearne regresije najmanjih kvadrata na dijelovima funkcije.

Također, UMAP koristi binarnu unakrsnu entropiju (3.14) kao funkciju gubitka umjesto KL divergencije kao što to radi t-SNE.

$$\begin{aligned} CE(X,Y) = \sum_i \sum_j \Bigg[& p_{ij}(X) \log \left(\frac{p_{ij}(X)}{q_i j(Y)} \right) \\ & + (1 - p_{ij}(X)) \log \left(\frac{1 - p_{ij}(X)}{1 - q_{ij}(Y)} \right) \Bigg] \end{aligned} \tag{3.14}$$

Laički rečeno, prvi član unutar sume omogućuje preslikavanje grupa ispravno, dok drugi član omogućuje preslikavanje razmaka ispravno zbog čega UMAP može opisati globalnu strukturu.

Konačno, umjesto gradijentog spusta (GD), UMAP koristi stohastički gradijentni spust (SGD) što ujedno ubrzava vrijeme izvođenja i zahtijeva manje memorije.

3.4. Usporedba algoritama za učenje mnogostrukosti

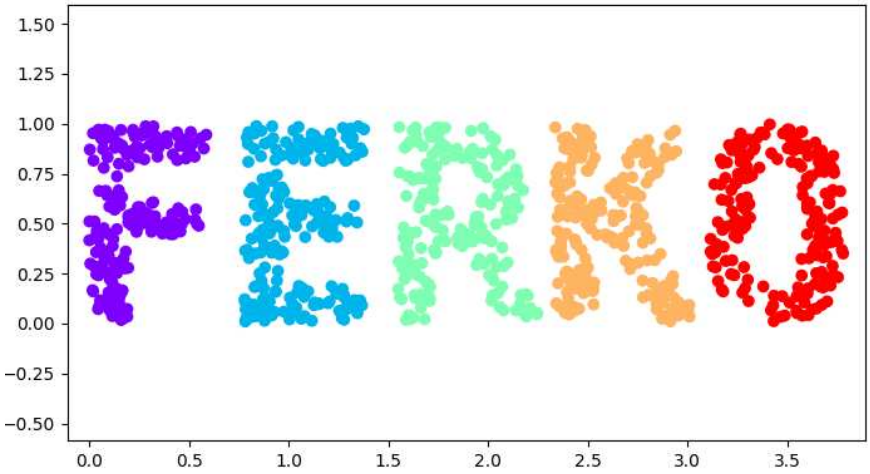
Iako matematika priča zanimljivu priču o navedenim algoritmima, kako bi se najbolje ilustrirale njihove sličnosti, različitosti i granice u ovom poglavlju uspoređivat će se kako pojedini algoritmi rade na određenim primjerima.

U prvom djelu poglavlja bit će napravljen jednostavan ilustrativni primjer. Kako bi se to izvelo, definiranje mnogostrukosti će biti napravljeno suprotno od zadatka učenja mnogostrukosti, točnije prvo će se definirati mnogostrukost u dvodimenzionalnom prostoru, a zatim će se ona ugraditi u trodimenzionalni prostor. Na ovaj način od algoritama za smanjenje dimenzionalnosti imamo dobro definiran cilj; "poništiti" ugradnju, tj. želimo da rezultat smanjenja bude originalna mnogostrukost. Ugrađivanje će biti izvedeno koristeći linearnu i nelinearnu transformaciju kako bi se pokazale razne težine ugrađivanja.

Originalna dvodimenzionalna mnogostrukost (vidljiva na slici 3.12) definirana je podacima u obliku riječi „FERKO”, gdje će svako slovo dobiti svoju zasebnu klasu (to će biti prikazano pomoću boje). Taj primjer odabran je kako bi se lako vidjelo bilo kakvo izobličenje te da se rotacije i zrcaljenja jasno vide.

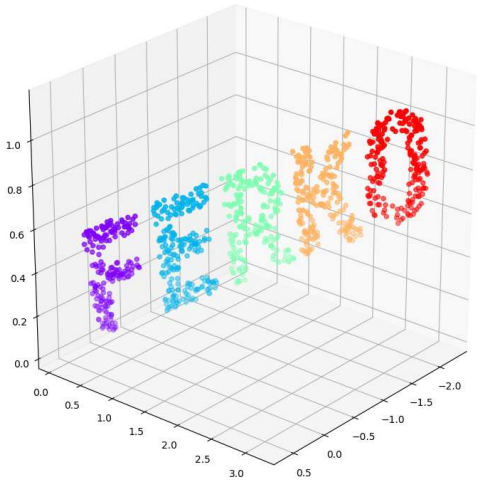
Također, radi provjere pristranosti algoritama, jedan primjer koji će biti korišten jesu slučajno generirani podaci.

Drugi dio poglavlja prikazat će usporedbu primjera na stvarnom skupu podataka, gdje mnogostrukost unaprijed nije poznata. Skup korišten u ovom radu je skup modnih artikala "Fashion-MNIST" koji se sastoji od crno bijelih slika veličine 28x28 razvrstanih u 10 razreda, svaki za jedan artikal. Broj susjeda koji koristimo bit će 100 za sve primjere.



Slika 3.12: Dvodimenzionalna mnogostrukost

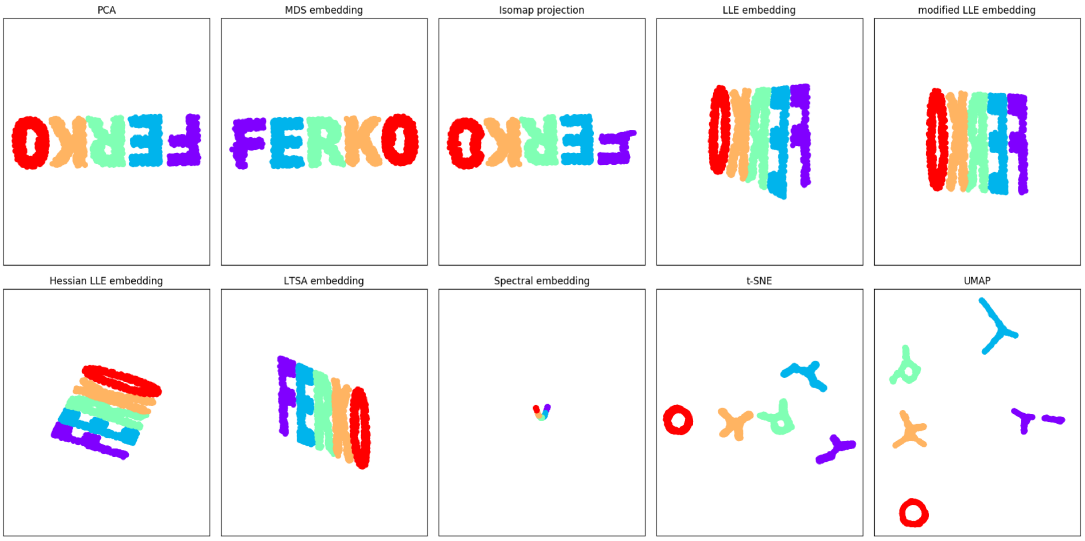
Primjer 1. Prvi primjer korišten u ovom radu je spomenuta mnogostrukost sa slike 3.12, ali ugrađena u trodimenzionalni prostor koristeći linearnu transformaciju. To je postignuto projiciranjem mnogostrukosti koristeći metodu slučajne projekcije. Rezultantna mnogostrukost prikazana je na slici 3.13



Slika 3.13: Mnogostrukost ugrađena u trodimenzionalni prostor

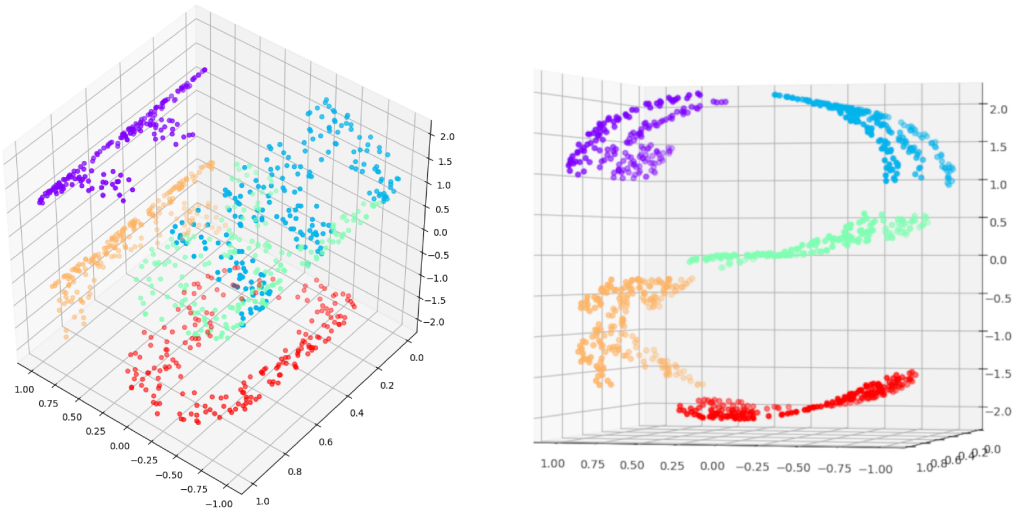
Na ovako jednostavnom primjeru očekivano je da svi algoritmi osim algoritama t-SNE i UMAP uspješno pronađu mnogostrukost. Zašto ne t-SNE i UMAP? Oni se razlikuju od ostalih algoritama za učenje mnogostrukosti tako što ne pokušavaju zadržati topologiju, već svojstva grupa. Iz tog razloga za očekivati je da će t-SNE i UMAP grupirati točke u zasebne grupe, svaku za svoje slovo, ali da će i dalje udaljenosti slova ostati slične. Dakle, da će slovo F biti najdalje slovu O (točnije da će ljubičasta grupa biti najdalje crvenoj grupi). Rezultati za sve algoritme vidljivi su na slici 3.14.

Kao što je bilo očekivano većina algoritama uspješno je naučila mnogostrukost na do skaliranje, rotaciju i zrcaljenje u ovom primjeru, mogu se primijetiti samo mali defekti na slovu F kod Isomap algoritma. No, vidljivo je da spektralno ugrađivanje (grupiranje), t-SNE i UMAP nisu rekonstruirali originalnu mnogostrukost. To je zato što su ti algoritmi namijenjeni za pronalaženje grupa u visokodimenzionalnom prostoru i preslikavanju tih grupa u niskodimenzionalni. Ali, kod algoritama t-SNE i UMAP može se primijetiti da su zadržani odnosi između grupa. Kao što je prije bilo očekivano grupe koje su prije bile topološki najdalje one to i dalje jesu, tj. može se vidjeti da razredi i dalje idu istim redoslijedom (ljubičasta do crvena), ali ne nužno u istom smjeru. Također, kod algoritama t-SNE i UMAP postoje oblici koji donekle liče na originalna slova.



Slika 3.14: Prikaz rezultata algoritama za učenje mnogostrukosti na linearnom primjeru

Primjer 2. Prošli primjer bio je vrlo jednostavan linearan primjer, no kako bi uspješno bile demonstrirane prednosti nekih algoritama nad drugima potrebno je pogledati malo složeniji primjer. Iz tog razloga, drugi primjer koristi mnogostrukost "savinutu" u oblik slova "S" i takvu ugrađuje u trodimenzionalni prostor. Ovako definirana mnogostrukost vizualizirana je na slici 3.15.



Slika 3.15: Mnogostrukost u obliku slova "S" ugrađena u trodimenzionalni prostor (iz dvije različite perspektive)

Mnogostrukost je postignuta transformacijom koju definiraju sljedeće relacije:

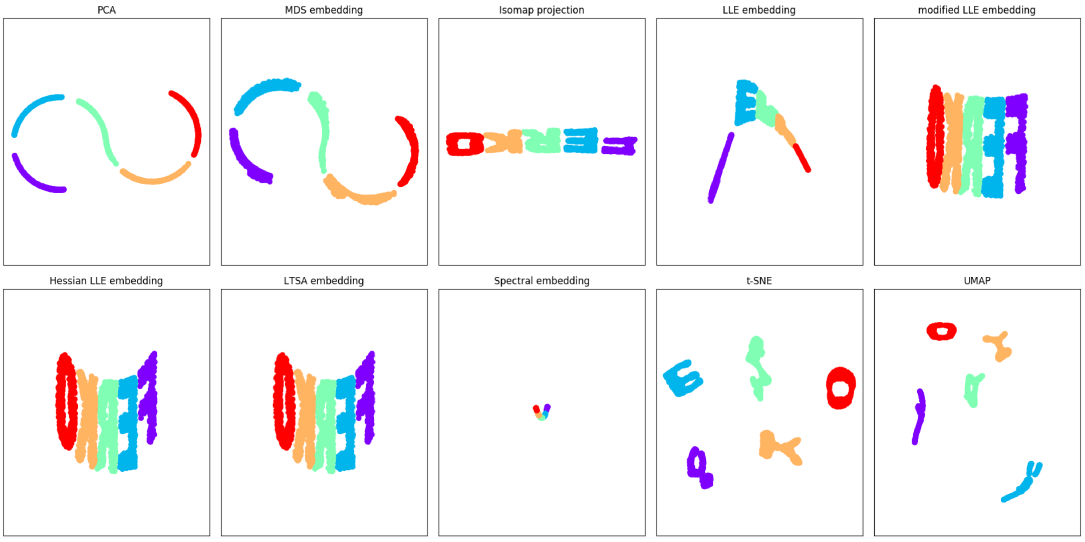
$$\vec{t} = 0.75\pi \cdot (\vec{x} - 2); \tag{3.15}$$

$$\vec{x'} = \sin(\vec{t}); \tag{3.16}$$

$$\vec{y'} = \vec{y}; \tag{3.17}$$

$$\vec{z'} = sign(\vec{t}) \cdot (\cos(\vec{t}) - 1) \tag{3.18}$$

Na ovakvom primjeru očekujemo da neće svi algoritmi uspješno pronaći ugrađenu mnogostrukost. Jedna od takvih metoda zasigurno je PCA koja zbog svoje linearne prirode ne može definirati odgovarajuću transformacijsku funkciju potrebnu za željeno mapirane u dvodimenzionalni prostor. Dodatno, očekivano je i za MDS da će imati poteškoće u transformaciji zbog korištenja euklidskih udaljenosti umjesto geodetskih što je vizualno opisano na slici 3.7. Za LLE je teško predvidjeti rezultat zbog velike ovisnosti algoritma o broju najbližih susjeda koji se koriste. Te nedostatke rješavaju algoritmi MLLE i HLLE od kojih, dakle, očekujemo da uspješno riješe ovaj problem. Očekivano ponašanje algoritma LTSA je slično onome od algoritma HLLE zbog matematičke sličnosti dviju metoda. Za spektralno ugrađivanje, t-SNE i UMAP ne bi trebala postojati velika razlika u pristupu ova dva primjera zbog čega očekujemo slične rezultate kao i prije.

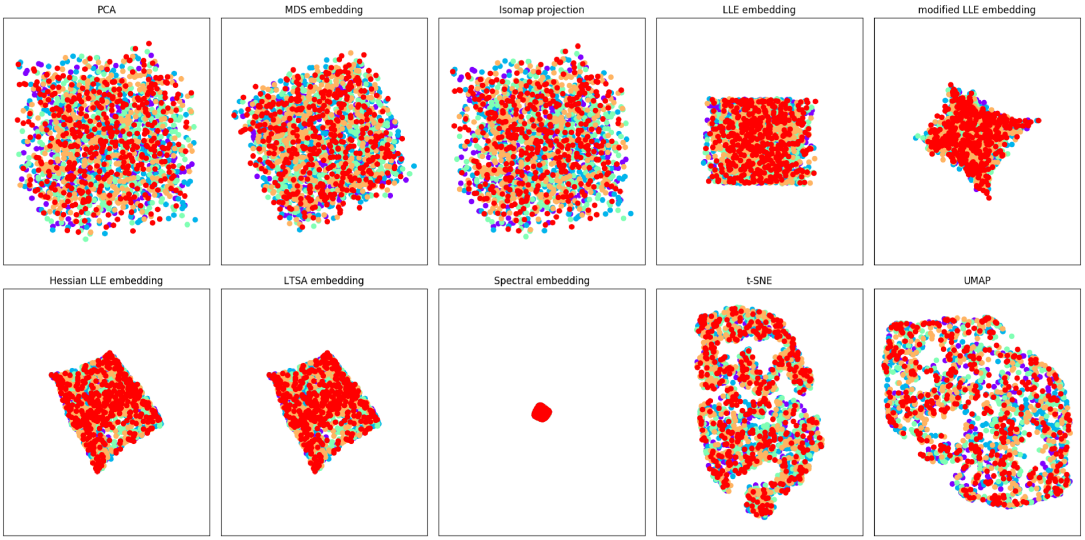


Slika 3.16: Prikaz rezultata algoritama za učenje mnogostrukosti na nelinearnom (S) primjeru

Iz ovog rezultata primjera (slika 3.16), za algoritam PCA možemo vidjeti da iako nije uspješno odredio ugrađenu mnogostrukost, uspješno je odredio smjerove dviju najvećih varijanci u podacima. Također su vidljivi nedostaci algoritma MDS. Kako

bi zadržao odnose euklidske udaljenosti MDS je odbacio y os. Isto tako vidljivo je značajno poboljšanje koje donose geodetske udaljenosti korištene u algoritmu Isomap. Problem regularizacije vrlo je očit u rezultatu algoritma LLE, ali vidljivo je i značajno poboljšanje koje donose modifikacije algoritama MLLE i HLLE. Algoritam LTSA očekivano je sličan algoritmu HLLE i očekivano je uspješan u ovom problemu. Spektralno ugrađivanje, t-SNE i UMAP imaju gotovo jednake rezultate kakvi su bili na prošlom primjeru, što je bilo i za očekivati.

Primjer 3. S ciljem utvrđivanja postoji li pristranost u algoritmima, dobra je praska provjeriti kako se ti algoritmi ponašaju na slučajno generiranim podacima. Naravno, za ovakav primjer očekujemo da algoritmi ne pronađu nikakvu ugrađenu mnogostrukost te da rezultantni nižedimenzionalni prostor izgleda jednako slučajan. Algoritmi koji održavaju svojstva grupa, kao što su spektralno ugrađivanje, t-SNE i UMAP mogli bi unijeti svoju pristranost prema grupiranju točaka. Rezultati ovog primjera vidljivi su na slici 3.17.



Slika 3.17: Prikaz rezultata algoritama za učenje mnogostrukosti na primjernu slučajno generiranih podataka

Iz rezultata, moguće je vidjeti kako većina algoritama ne unosi mnogo pristranosti. Algoritam spektralnog ugrađivanja svrstao je podatke u jednu grupu, ali zato algoritmi t-SNE i UMAP, usprkos nepostojećoj temeljnoj strukturi forsiraju stvaranje grupa. Dodatno, valja naglasiti da je ovaj problem više istaknut kod t-SNE algoritma.

Skup podataka "Fashion MNIST" Fashion MNIST je skup podataka odjevnih predmeta kompanije Zalando, a dostupan je na github.com/zalandoresearch/fashion-mnist. Sastoji od skupa za učenje koji sadrži 60000 primjera i od skupa za testiranje koji sadrži dodatnih 10000 primjera. Svaki primjer je crno-bijela slika odjevnog artikla veličine 28x28 (tj. ima 784 značajke) grupirane u 10 razreda. Primjeri slika kakve nalazimo u ovom skupu podataka mogu se vidjeti na slici 3.18. Ideja skupa podataka Fashion MNIST je da bude direktna zamjena za MNIST; skup rukom pisanih znamenaka koji je vrlo popularan u svijetu strojnog učenja. Iz tog razloga skup Fashion MNIST dizajniran je da ima jednaku veličinu slika i strukturu skupova za učenje i testiranje kao i MNIST.

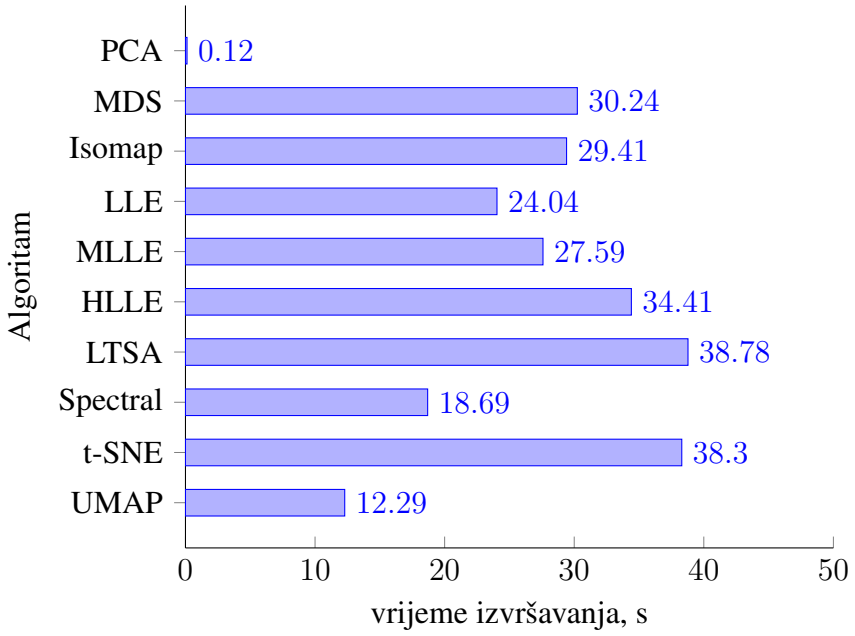
Slika 3.18: Dio skupa podataka Fashion MNIST

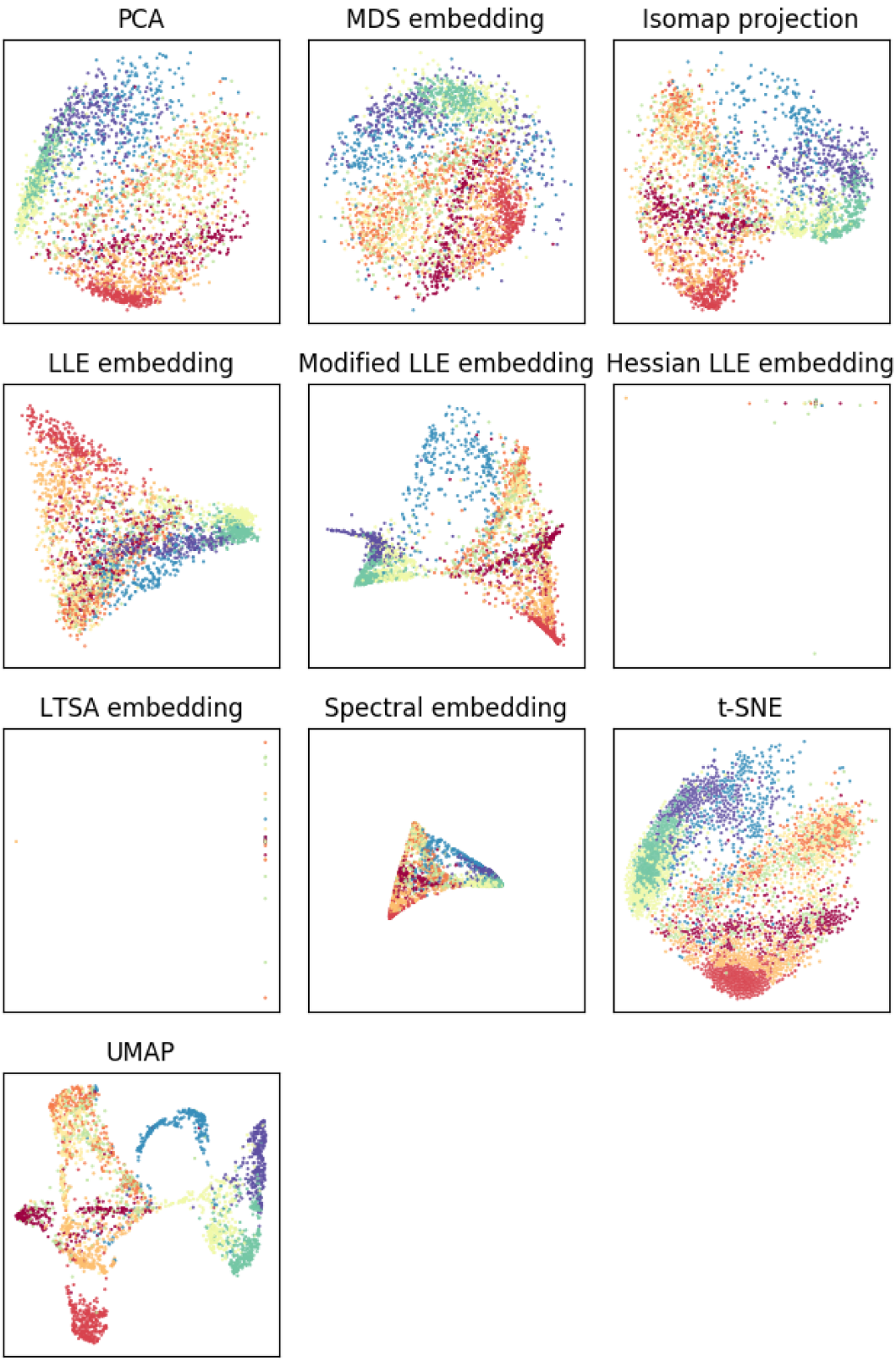
S obzirom na postojanje mnogih primjera korištenja algoritama za smanjenje dimenzionalnosti na skupu podataka MNIST, jedan od kojih je Pedregosa et al. (2011), u ovom radu odlučeno je pokazati kako ovi algoritmi funkcioniraju na zahtjevnijem skupu podataka kao što je Fashion MNIST koji predstavlja problem koji je bliži onima iz stvarnog svijeta.

Na slici 3.19 vidljivi su vrlo zanimljivi rezultati koji pokazuju ponašanje na primjeru sličnijem stvarnim podacima. Cijelo vrijeme tijekom evaluiranja rezultata bitno je imati na umu kako je izvedeno smanjenje dimenzionalnosti generirala iznimno niskodimenzionalnu mnogostrukost od samo dvije dimenzije. Ovo je prvenstveno zbog ograničenja mogućnosti vizualiziranja viših dimenzija kvalitetno. No, uzevši to u obzir, ideja ovog primjera je pokazati neka od svojstava ovih algoritama.

Iz ovog se primjera može vidjeti iako na jednostavnim primjerima algoritmi rade vrlo dobro, u stvarnosti situacija je puno kompliciranа. Algoritam PCA, iako naravno nesavršen, pokazuje zanimljive rezultate vrlo slične onima od kompliciranijih algoritmima, kao što je MDS, u vremenu koje je nekoliko stupnjeva manje od svih ostalih algoritama. U rezultantnim mnogostrukostima tih algoritama možemo vidjeti odvajanje razreda do neke mjere, ali neki razredi koji su prostorno sličniji drugima su se pomiješali. Marginalno bolji rezultat dobiven je projekcijom Isomap. Algoritmi HLLE i LTSA, očekivano, dijele vrlo slične rezultate, oba koja su nažalost neupotrebljiva. Naravno, to ne znači da su ovi algoritmi loši, daleko od toga, to samo znači da na ovom konkretnom primjeru (podacima i rezultantnoj dimenziji) oni nisu uspješno mogli odrediti ugrađenu mnogostrukost. Izuzev njih, LLE i algoritam spektralnog ugrađivanja rezultirali su najlošijom vizualizacijom te je miješanje grupa kod njih najizraženije. Nedostatke algoritma LLE kod ovog primjera popravio je algoritam MLLE kod kojeg se grupiranje može jasno vidjeti. Algoritam t-SNE, koji je od svog izlaza bio standard za ovakvu primjenu, definitivno je rezultirao vizualizacijom koja je lošija od očekivanog. Naravno, ukoliko bi se parametar *preplexity* povećao sa 100 na 200 rezultantna slika bi pokazivala bolje grupiranje, ali bi to ujedno unijelo ogroman vremenski trošak na već ovako zahtjevan algoritam. Uz to, t-SNE pokazuje svojstva grupiranja koja su bolja nego prijašnji algoritmi. Zadnji, ali ne manje važan, algoritam UMAP dao je objektivno najbolje rezultate, s jasnim granicama većine grupa, u vremenu koje je kraće od svih nelinearnih algoritama testiranih u ovom radu.

Trajanje izvođenja algoritama na skupu Fashion MNIST





Slika 3.19: Rezultati algoritama na skupu Fashion MNIST

4. Primjena smanjenja dimenzionalnosti u SCA

Kao što je spomenuto u uvodu, ovaj rad je temeljen na radu od Picek et al. (2019) u kojem se autori koncentriraju na utjecaj tehnika odabira značajki (engl. *feature selection*) u analizi koja koristi sporedna sredstva. Točnije, cilj je odrediti kako efikasnost SCA može biti povećana korištenjem tehnike odabira značajki. Dodatno, u radu je istražena i metoda PCA kako bi se dao uvid u razlike odabira značajki i smanjenja dimenzionalnosti. Iz spomenutog rada preuzeto je označavanje i skupovi podataka korišteni za evaluaciju metoda. U ovom poglavlju bit će opisana i raspravljena primjena smanjenja dimenzionalnosti u analizi koja koristi sporedna sredstva uređaja. Radi boljeg razumijevanja dodatno će se opisati bitniji koncepti korišteni prilikom metode prezentirane u radu koji nisu obuhvaćeni prijašnjim poglavljima. Zatim će biti opisana metoda kojom su dobiveni rezultati, a u posljednjem odjeljku ovog poglavlja ti dobiveni rezultati bit će izneseni, vizualizirani i raspravljeni.

4.1. Opis problema

4.1.1. Kriptografski algoritam AES

Američki nacionalni institut standarda i tehnologije (engl. *National Institute of Standards and Technology, NIST*) je u razdoblju od 1997. do 2000. godine raspisao natječaj s ciljem pronalaska novog standarda u kriptografiji koji bi naslijedio DES. Taj novi standard bi se zvao napredni enkripcijski standard (engl. *Advanced Encryption Standard, AES*). Od mnoštva kandidata petnaest je odabrano u uži krug, a od njih zatim pet finalista: Rijndael, Serpent, MARS, Twofish, RC6. Iako su svi finalisti imali jednaku jačinu sigurnosti, NIST je odabrao algoritam Rijndael kao novi standard zbog dobrih performansi na većini platformi te jednostavnosti implementacije. Rijndael je simetrični blokovski kriptoalgoritam koji su razvili dvojica belgijskih kriptografa Joan

Daemen i Vincent Rijmen, po kojima je i imenovan. Rijndael može biti specificiran s veličinom bloka i ključa koja je višekratnik od 32 bita između 128 bita i 256 bita. S druge strane, AES ima fiksnu veličinu bloka od 128 bita i veličinu ključa od 128, 192 i 256 bita. AES je, dakle, podskup Rijndael algoritma. Broj krugova T u AES-u ovisi o veličini ključa te za verzije AES-128, AES-129 i AES-256 je redom 10, 12 i 14. AES koristi četiri operacije: $SubBytes$, $ShiftRows$, $MixColumns$ i $AddRoundKey$. Funkciju jednog kruga, $R_i, 1 \leq i \leq T$ definiramo kao

$$R_i = \begin{cases} AddRoundKey_i \circ MixColumns \circ ShiftRows \circ SubBytes & , i < T \\ AddRoundKey_i \circ ShiftRows \circ SubBytes & , i = T \end{cases} \tag{4.1}$$

Prije prvog kruga, u koraku $AddRoundKey_0$ koristi se izbijeljeni ključ tako da se enkripcija sa T krugova i ključem K opisuje kao

$$E_k = R_T \circ \cdots \circ R_1 \circ AddRoundKey_0. \tag{4.2}$$

Sve četiri operacije izvode se na 128 bitnom bloku raspoređenom u matricu 4×4 oblika

$$\begin{pmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{pmatrix}$$

i na Rijndaelovom konačnom polju $\mathbb{F}_{256}$.

Operacija $SubBytes$: U ovoj operaciji svaki od 16 bajtova u matrici je zamijenjen s drugom vrijednošću prema tablici koja se naziva supstitucijska kutija (engl. *Substitution box*, dalje S-Box). Ova tablica je javna, a dizajnirana je tako da nije moguće da se vrijednost preslikava sama u sebe te nije moguće da se svi bitovi preokrenu.

Operacija $ShiftRows$: Ova operacija je vrlo jednostavna te sve što radi je za i-ti red tablice, gdje za i vrijedi $0 \leq i \leq 3$, cirkularno rotira vrijednosti lijevo za i pozicija.

Operacija $MixColumns$: Ova operacija nastavlja se na prošlu te se u njoj sada miješaju stupci. Ovaj korak se izvodi pomoću matričnog množenja. Dakle, svaki stupac, kao vektor se množi s matricom

$$M = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}$$

Prilikom dekripcije koristi se inverz matrice M^{-1}.

Operacija $AddRoundKey$**:** Konačno, u ovoj se operaciji našoj matrici dodaje ključ trenutnog kruga K_i koristeći operaciju isključivo-ILI.

4.1.2. Slučajna šuma

Slučajna šuma (engl. *Random forest*, skrać RF) vrlo su poznat stohastički ansambl osnovnih klasifikatora stabala odluke.

Stabla odluke su neparametarska, diskriminativna metoda aproksimacije funkcije diskretnih vrijednosti koja može učiti disjunktivne koncepte i robusna je na šum. Stablo odluke koristi vrlo intuitivan koncept, a sastoji se od čvorova koji predstavljaju pitanja, a grane čvorova predstavljaju odgovore na ta pitanja. Listovi stabla su konačni razredi. Dakle, na nekom čvoru se "odgovara na pitanje", a odgovor na to pitanje zatim vodi grananjem na neko drugo pitanje. Odabir atributa koji će činiti čvorove izvodi se koristeći informacijsku dobit, koja je usko vezana s entropijom. Korijeni stabla su atributi s najvećom informacijskom dobiti.

Definicija 4 *Za neku diskretnu slučajnu varijablu X s mogućim skupom vrijednosti $\{x_1, \ldots, x_n\}$ i funkciju vjerojatnosti P definiramo entropiju kao:*

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_b P(x_i) \tag{4.3}$$

Definicija 5 *Neka je $\boldsymbol{D} = \left\{(\boldsymbol{x}^{(i)}, y^{(i)})\right\}_{i=1}^{N}$ skup označenih primjera, gdje je $x_a \in \mathcal{V}(a)$ je vrijednost a-tog atributa primjera $\boldsymbol{x}^{(i)}$, a y oznaka odgovarajućeg razreda. H je entropija definirana u definiciji 4. Za vrijednost v i atribut a neka*

$$S_a(v) = \{\boldsymbol{x} \in \boldsymbol{D} | x_a = v\} \tag{4.4}$$

bude definiran kao podskup od $\boldsymbol{D}$ za koji atribut a jednak vrijednosti v. Tada informacijsku dobit (engl. gain*) skupa $\boldsymbol{D}$ za atribut a definiramo kao*

$$IG(\boldsymbol{D}, a) = H(D) - \sum_{v \in \mathcal{V}(a)} \frac{|S_a(v)|}{|\boldsymbol{D}|} H(S_a(v)) \tag{4.5}$$

Slučajne šume sastoje se od mnoštva ovakvih stabla odluke, svako od kojih daje zasebnu klasifikaciju, a zatim razred s najviše "glasova" postane rezultantan razred

modela RF. Kako bi se osnovni klasifikatori maksimalno nadopunjavali, potrebno je da imaju vrlo malu međusobnu korelaciju. Kako bi se to postiglo, iskorištava se svojstvo stabla odluke da bude vrlo osjetljivo na podatke na kojima je učeno, tako da se za svako stablo slučajno iz skupa podataka uzorkuje s ponavljanjem. Ovaj postupak naziva se *agregiranje bootstrap* (engl. *bagging*). Vremenska kompleksnost učenja ovog algoritma je $O(I \cdot k \cdot N \cdot \log N)$ gdje je I broj stabala u slučajnoj šumi, k je broj atributa koji se razmatra u svakom čvoru stabla, a N je broj uzoraka.

4.1.3. Entropija pogađanja

Entropija pogađanja (engl. *guessing entropy*, dalje GE), koju je originalno definirao Massey (1994), predložena je za korištenje u analizi koja koristi sporedna sredstva od Köpf i Basin (2007). Ova mjera jednostavno predstavlja težinu da napadač pogodi vrijednost slučajne varijable. Točnije, GE neke slučajne varijable X je prosječan broj pitanja tipa "vrijedi li da je $X = x$" koja moraju biti postavljena da bi se točno odredila vrijednost X. Ako pretpostavimo da su vjerojatnosti p poznate, optimalna procedura je pokušati sve vrijednosti x prema opadajućem redoslijedu njihovih vjerojatnosti. Generalnije, neka je $\mathcal{X}$ indeksiran tako da uvijek vrijedi $p_x(x_i) \geq p_x(x_j) \iff i \leq j$. Tada se entropija pogađanja od X, $G(X)$, definira kao

$$G(X) = \sum_{1 \leq i \leq |\mathcal{X}} i p_x(x_i) \tag{4.6}$$

Analogno uvjetnoj entropiji, uvjetna entropija pogađanja definira se kao

$$G(X|Y) = \sum_{y \in \mathcal{Y}} p_y(y) G(X|Y = y) \tag{4.7}$$

koja predstavlja broj pogađanja potrebnih za određivanje vrijednosti X kada je vrijednost Y poznata. Dakle, možemo povezati entropiju pogađanja s pojmom dobrobiti koji se koristi u kontekstu višestruke linearne kriptoanalize kao mjera koliko je kompleksnost iscrpnog pretraživanja smanjenja zahvaljujući napadu (Standaert et al., 2009).

4.2. Skupovi podataka

U ovom radu, korištena su tri skupa podataka, koja predstavljaju razne težine situacija koje se često mogu dogoditi. Prvi, najjednostavniji, skup podataka sadrži podatke koji ne implementiraju protumjere osim maskiranja te koji sadrže vrlo malo šuma. Drugi, malo realniji, primjer je skup podataka koji kao ni prvi nema implementirane protumjere osim maskiranja, ali zato ima mnogo šuma u podacima. Ovakva situacija češća

je u stvarnosti zbog nemogućnosti izvođenja savršenih mjerenja sporednih sredstava. Konačno, posljednji skup podataka koji je korišten u ovom radi, implementira protumjere u obliku slučajnih kašnjenja. Kao što je već spomenuto, ovi skupovi podataka opisani su u Picek et al. (2019).

DPAcontest v4

U ovom skupu podataka sadržana su mjerenja maskirane programske implementacije AES. Kako je maska poznata, lagano je pretvoriti ove podatke u skup bez protumjera. Budući da je ovo programska implementacija, većina odljeva informacija događa se za vrijeme izvođenja zamjene pomoću kutija za zamjene (engl. *Substitution box*, dalje S-Box), a ne za vrijeme zapisivanja podataka u registre. Zbog toga se napad izvodi u prvom krugu, a model odljeva je definiran kao

$$Y(k*) = \text{S-box}\left[P_{b_1} \oplus k*\right] \oplus \underbrace{M}_{\text{poznata maska}}, \tag{4.8}$$

gdje je p_{b_1} b_1-ti oktet originalne poruke. U ovom radu napada se samo prvi oktet, dakle $b_1 = 1$.

AES_HD

Ovaj skup podataka predstavlja nezaštićenu implementaciju AES-128 koja je napisana u VHDL-u. Za eksternu komunikaciju koristi se modul UART, dok je za implementaciju korišten Xilinxov Virtex-5 FPGA na evaluacijskoj ploči SESEBO GII. Sporedna sredstva mjerena su usko-poljnom elektromagnetskom sondom visoke osjetljivosti na kondenzatoru za odvajanje smještenom na krugu napajanja. Mjerenja su uzorkovana na osciloskopu Teledyne LeCroy Waverunner 610zi. Ukupni obujam dizajna je 1850 tablica traženja (engl. *lookup tables*) i 742 bistabila. U ovom slučaju napad se radi na posljednji krug AES-a tako da je prikladan i često korišten model odljeva onaj koji opisuje zapisivanje podataka u registre prilikom zadnjeg kruga, tj.

$$Y(k*) = HW\Big(\underbrace{\text{S-box}^{-1}\left[C_{b_1} \oplus k*\right]}_{\text{prošla vrijednost registra}} \oplus \underbrace{C_{b_2}}_{\text{oktet kriptirane poruke}}\Big), \tag{4.9}$$

Random delay

Posljednji slučaj je stvarna zaštićena programska implementacija AES-a izvedena na 8-bitnom Atmelovom AVR mikrokontroleru. Implementirana zaštita je u obliku slučajnih kašnjenja. Napad se kao i u prvom skupu podataka izvodi na prvu operaciju S-box.

4.3. Metoda

Za sve korištene skupove podataka, primijenjena su dva različita modela odljeva. Osnovniji koristeći direktno međuvrijednosti tragova i drugi koji koristi Hammingovu težinu HW. Ti modeli su označeni redom kao "value" i "HW". Broj značajki D za primjer iz svakog skupa odabran je da bude 50, pri čemu je 50 značajki dobiveno kao najbolje rangirane korištenjem Pearsonove korelacije s modelom. Iz inicijalnog skupa podataka napravljena su dva podskupa: podskup podataka $\boldsymbol{D}_{train}$ za učenje koji sadrži 10 000 tragova i podskup za testiranje $\boldsymbol{D}_{test}$ koji sadrži 25 000 tragova. Valja napomenuti da vrijedi $\boldsymbol{D}_{train} \cap \boldsymbol{D}_{train} = \emptyset$. Cilj je za skup podataka dimenzije D smanjenjem dimenzionalnosti pronaći skup podataka dimenzije D_m gdje vrijedi da je $D_m < D$. Metode smanjenja dimenzionalnosti koje se koriste su one opisane u prošlim poglavljima[1]. Za većinu metoda potrebno je odrediti broj susjeda N_n koji algoritam koristi prilikom smanjenja. Eksperimentalno izabrane kombinacije parametara korištene D_m i N_n korištene u ovom radu uz označavanje

$$(D_m, A) = \{(D_m, N_n) | N_n \in A\}, \quad D_m \in \mathbb{N}, A \subseteq \mathbb{N}$$

mogu biti opisane kao

$$S = \{(3, \{10\}) \cup (7, \{10, 20, 40\}) \cup (10, \{10, 30, 70\}) \cup (15, \{10, 50, 70, 150\}) \cup \\ (25, \{10, 50, 400\}) \cup (40, \{50, 500, 1000\})\}$$

Za neke algoritme, neke kombinacije navedenih parametara nisu moguće te su one zanemarene. Dodatno je korišten trivijalni transformator definiran funkcijom identiteta $f(x) = x, \forall x$, koji predstavlja slučaj u kojemu prije klasifikacije ne dolazi do transformacije originalnih podataka, sa ciljem uspostavljanja referentne vrijednosti. On je definiran samo radi generalizacije procesa izvođenja eksperimenata, a označen je pseudonimom "dummy". Uz osnovne metode smanjenja dimenzionalnosti dodana je i hibridna metoda koja koristi rijetke slučajne projekcije (engl. *Sparse Random Projection, SRP*). Za korištene skupove podataka pokazalo se da SRP u dimenziju 30 zadržava razumnu količinu informacija dok istovremeno pruža poželjnu razinu smanjenja. Zato hibridne metode prvo, koristeći metodu SRP, reduciraju dimenzionalnost u dimenziju 30, a zatim korištenjem neke osnovnih metoda smanjenje dimenzionalnosti se prostor dodatno smanjuje u dimenziju D_m za koju vrijedi $D_m < 30$. Takve hibridne metode notirane su koristeći prefix "srp30+" uz naziv osnovne metode smanjenja korištene.

[1] Algoritmi MDS i t-SNE nisu korišteni u ovom dijelu rada zbog nemogućnosti generalizacije na neviđenim primjerima

Nakon transformacije, skup podataka koji sada ima dimenziju D_m postavlja se kao ulaz klasifikatora. Klasifikator korišten u ovom radu je opisana metoda slučajnih šuma koji je odabran zbog otpornosti na šum što je u ovom radu od velike važnosti. Radi usporedbe metoda za smanjenje dimenzionalnosti broj stabala korištenih u ovom koraku je fiksno postavljen na 100. Naravno, za određenu metodu i njene parametre moguće je pronaći optimalne parametre prilikom klasifikacije. Ako koristimo model odljeva s međuvrijednosti(value) broj razreda je 256 te predstavlja dekadsku vrijednost okteta koji pokušavamo odrediti, dok ako je riječ o metodi s Hammingovom težinom tada je broj razreda 9 i on naravno odgovara Hammingovoj težini okteta. Kao mjeru performansi koristi se točnost klasifikatora, no uz to, zbog činjenice da ta mjera često ne pokazuje stvarnu situaciju, dodatno se koristi stopa uspjeha (engl. *success rate*) i entropija pogađanja.

4.4. Rezultati

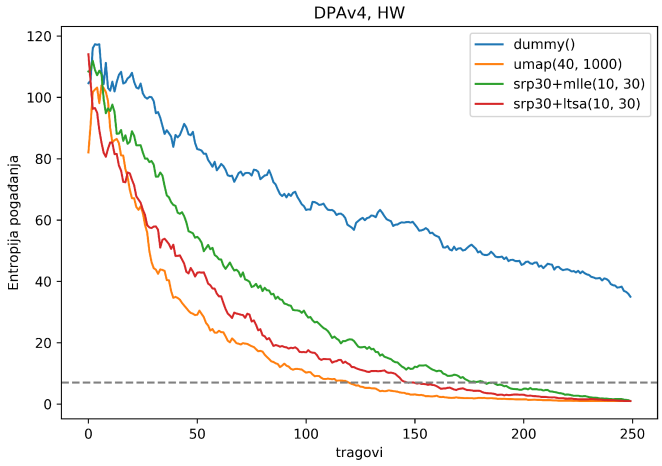
U tablicama 4.1-4.6 prezentirane su vrijednosti točnosti klasifikatora i minimalna dobivena vrijednost entropije pogađanja GE_{min}, a na slikama 4.1-4.6 prikazane su vrijednosti entropije pogađanja ovisne o broju tragova. Zbog prostornih ograničenja, prikazani podaci predstavljaju reprezentativni podskup testiranih scenarija. Rezultati svih izvedenih testiranja dostupni su na https://github.com/eugen-vusak/manifold-SCA/blob/master/practical/reports/summary.xlsx. Za tablice 4.1-4.6 uzete su vrijednosti deset najboljih metoda ne uključujući metodu "dummy", koja je zatim dodana zasebno. Na isti način, za slike 4.1-4.6 su odabrane tri najbolje vrijednosti (bez metode "dummy"), i dodatno referentna metoda "dummy". Na navedenim slikama isprekidanom sivom linijom označena je granica za koju smatramo da je napad bio uspješan, a metode su u legendi opisane oblikom "ime_metode(D_m, N_n)", iako ako metoda ne koristi neke parametre oni su izostavljeni i iz naziva.

4.4.1. DPAcontest v4

Tablice 4.1 i 4.2 pokazuju rezultate za skup podataka DPAcontest v4 dobivene za modele odljeva HW i value. Za model HW na tablici 4.1 možemo vidjeti kako je najbolja metoda UMAP s vrlo malim smanjenjem za samo 10 dimenzija. U kontrastu s time druge dvije najuspješnije metode su hibridne metode kod kojih je u oba slučaja rezultantna dimenzija 10. Te metode koriste metode LTSA i MLLE kao osnovnu metodu. Metoda LLE pokazala se relativno uspješnom s raznim rasponom parametara. Bitno

je naglasiti da su ove metode postigle mnogo bolje rezultate od referentne metode. Za ovaj specifični eksperiment pokazalo se da dimenzija 10 s 30 najbližih susjeda globalno pokazuje najbolje rezultate. Također, vidljivo je da visoka klasifikacijska točnost ne rezultira nužno dobrim rezultatima u kontekstu entropije pogađanja. Na slici 4.1 vidljivo je da je razlika između prve tri metode relativno velika te da ovisno o metodi broj potrebnih tragova može odstupati za približno 40.

Rezultati za odljev value, vidljivi na tablici 4.2 prikazuju drugačiju sliku. U ovom slučaju dvije najbolje metode obje izvode smanjenje u prostor od 25 dimenzija. I u ovom slučaju metoda UMAP pokazala se najuspješnijom, ali u ovom slučaju to je s većim smanjenjem. Dodatno metoda srp30+UMAP pokazala se vrlo uspješnom, marginalno lošijom od metode MLLE. Slika 4.2 pokazuje kako je razlika između najboljih metoda puno manja nego u modelu HW.



Slika 4.1: GE za skup DPAcontest v4 – model HW

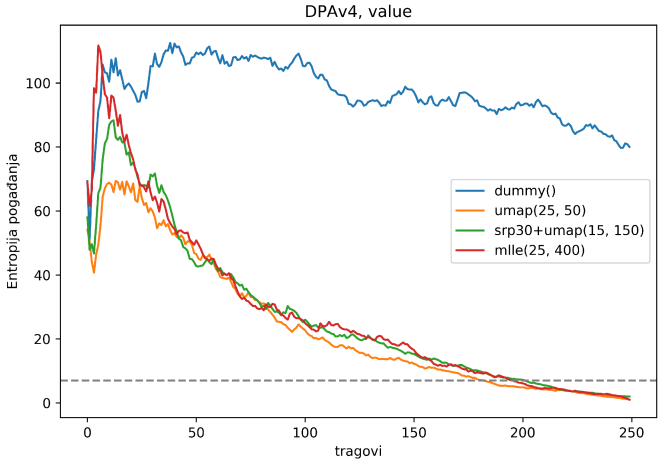
Tablica 4.1: Rezultati za skup DPAcontest v4 – model HW

metoda	D_m	N_n	točnost	GE_{min}
umap	40	1000	0.74448	1
srp30+ltsa	10	30	0.72476	1
srp30+mlle	10	30	0.79672	1
lle	10	30	0.73048	1.78
lle	25	200	0.84988	1.86
lle	20	50	0.7438	2
lle	25	400	0.86236	2
mlle	20	50	0.8826	2
srp30+mlle	3	10	0.71436	2
lle	40	500	0.83216	2.58
dummy			0.85656	35

4.4.2. AES_HD

Za skup podataka AES_HD rezultati su dani u tablicama 4.3 za model odljeva HW i u 4.4 za model odljeva value. Za model HW, iako su korištenjem metoda smanjenja dimenzionalnosti postignuti bolji rezultati nego kada one nisu implementirane, niti jedna od metoda nije uspješno izvela napad što je vidljivo na slici 4.4. Tome je najbliže bila metoda MLLE, dok su sve tri sljedeće metode hibridnog oblika raznih parametara. Manjak odnosa točnosti klasifikatora vrlo je izražena i u ovom primjeru.

Za rezultate na istom skupu podataka, ali koristeći model odljeva value, u rezultatima iz tablice 4.4 vidljivo kako hibridna metoda koja koristi UMAP, a traženi prostor je vrlo niskodimenzionalan prostor dimenzije 7, je rezultirala najboljim rezultatom.

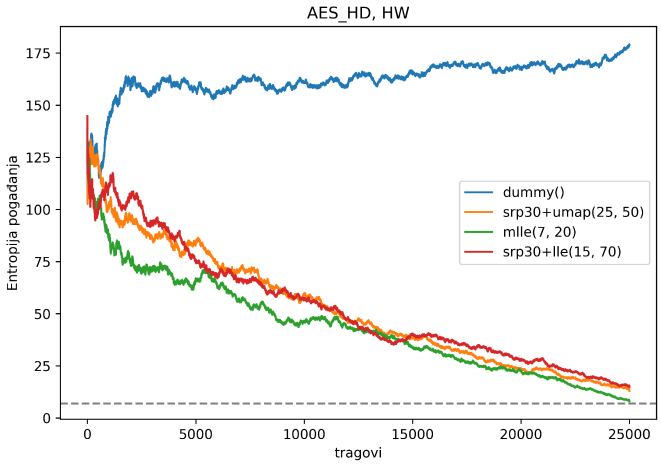


Slika 4.2: GE za skup DPAcontest v4 – value model

Tablica 4.2: Rezultati za skup DPAcontest v4 – value model

metoda	D_m	N_n	točnost	GE_{min}
umap	25	50	0.146	1
mlle	25	400	0.29872	1
srp30+umap	15	150	0.12028	2
srp30+hlle	3	10	0.00556	2
ltsa	7	20	0.00464	2.88
umap	40	1000	0.132	2.9
lle	3	10	0.12292	3
lle	3	10	0.1212	3
mlle	10	10	0.0044	3
srp30+mlle	10	10	0.00444	3
dummy			0.3466	50.98

Ista metoda, ali za konačnim prostorom dimenzije 25 je sljedeća. U ovom slučaju generalno se najbolje pokazala transformacija u 7-dimenzionalni prostor. Na slici 4.4 može se vidjeti nakon koliko tragova su metode smatrane uspješnima. Točnost na ovom skupu podataka vrlo je niska i nalikuje slučajnom pogađanju $(1/256)$. Za relativno mali broj primjera korišten u skupu za učenje, na ovako teškom skupu s 256 razreda ovakav rezultat nije daleko od očekivanog.



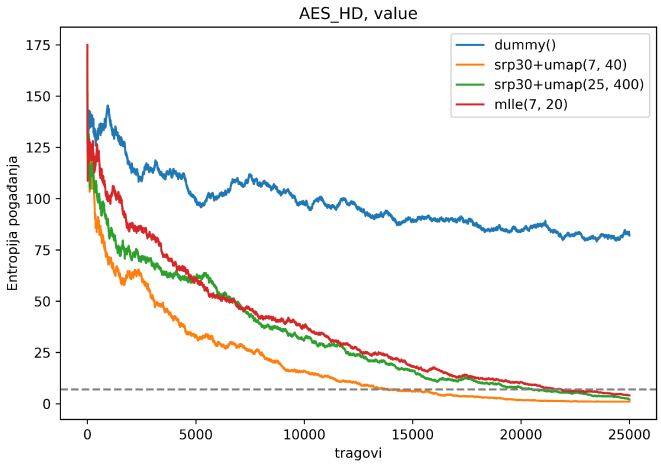
Slika 4.3: GE za skup AES_HD – model HW

Tablica 4.3: Rezultati za skup AES_HD – model HW

metoda	D_m	N_n	točnost	GE_{min}
mlle	7	20	0.23452	8
srp30+umap	25	50	0.05728	13
srp30+lle	15	70	0.25116	15
srp30+ltsa	10	70	0.22916	15.58
umap	25	400	0.01376	20.56
mlle	10	70	0.24068	21.96
umap	10	70	0.0436	22.68
srp30+umap	3	10	0.22524	23.48
mlle	15	150	0.24896	23.78
srp30+ltsa	15	50	0.22136	23.98
dummy			0.24904	103

4.4.3. Random delay

Konačno, rezultati za posljednji skup podataka, Random delay, koji predstavlja podatke najbliže stvarnosti, dani su u tablicama 4.5 i 4.6 redom za modele HW i value. Za model HW rezultati pokazuju da iako nakon mnogo tragova je moguće postići ni-



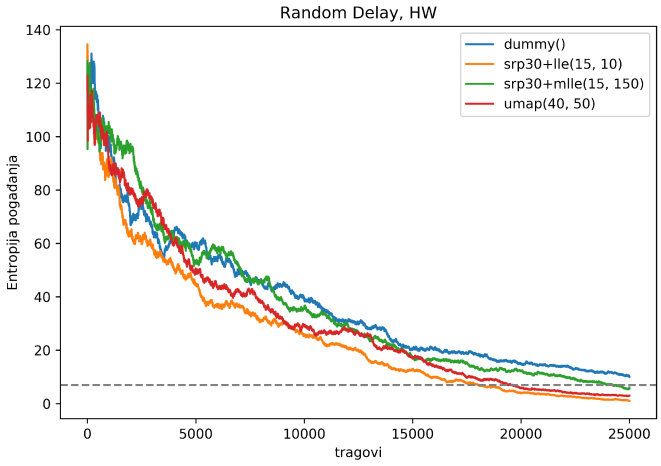
Slika 4.4: GE za skup AES_HD – value model

Tablica 4.4: Rezultati za skup AES_HD – value model

metoda	D_m	N_n	točnost	GE_{min}
srp30+umap	7	40	0.0038	1
srp30+umap	25	400	0.00364	2
mlle	7	20	0.00372	4
srp30+lle	10	10	0.00448	6.78
srp30+umap	7	20	0.00428	7
pca	3		0.00416	7.72
mlle	20	150	0.00424	7.76
srp30+mlle	15	50	0.00408	8.22
srp30+umap	7	10	0.004	10.96
lle	7	10	0.00388	11.06
dummy			0.00412	79.06

sku vrijednost GE, korištenje metoda smanjenja dimenzionalnosti nije postiglo veliku korist, naspram slučaja gdje te metode nisu korištene. Ovo je posebno izraženo na slici 4.5. U svakom slučaju, najboljom metodom u ovom slučaju pokazala se metoda srp30 + LLE, s konačnom dimenzijom od 15. Iza nje, kao i u primjeru DPAv4-HW, iskazala se metoda UMAP s malim smanjenjem dimenzionalnosti.

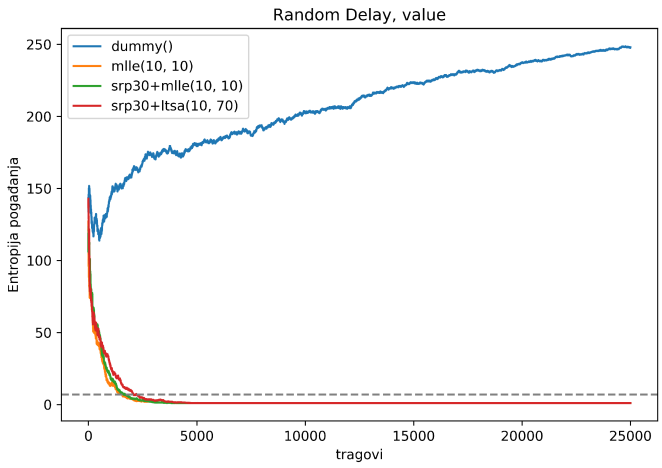
No, kod korištenja modela value, metodama smanjenja dimenzionalnosti postignuti su dobri rezultati izneseni u tablici 4.6. Dvije najbolje metode, gotovo ekvivalentnih rezultata, su metoda MLLE s parametrima (10,10) te inačica iste metode s predobradom koristeći metodu SRP. Za te metode napad je bio uspješan, tj. postigao je niski GE, nakon samo 2000 tragova. To je vidljivo na slici 4.6. Za ovu metodu generalno su najbolje rezultate postigle metode MLLE, HLLE i LTSA, s i bez dodanog koraka metode SRP, dok su metode LLE, PCA i UMAP pokazale lošije rezultate.



Slika 4.5: GE za skup Random Delay – model HW

Tablica 4.5: Rezultati za skup Random Delay – model HW

metoda	D_m	N_n	točnost	GE_{min}
srp30+lle	15	10	0.24632	1
umap	40	50	0.03268	2.74
srp30+mlle	15	150	0.252	5.34
srp30+lle	10	70	0.24924	5.38
ltsa	10	10	0.24428	6.8
mlle	7	10	0.2378	7
umap	7	40	0.04592	8.82
srp30+hlle	25	400	0.25472	9.74
dummy			0.25728	9.9
srp30+lle	3	10	0.2288	10.06
srp30+lle	7	40	0.24364	10.16



Slika 4.6: GE za skup Random Delay – value model

Tablica 4.6: Rezultati za skup Random Delay – value model

metoda	D_m	N_n	točnost	GE_{min}
srp30+mlle	10	10	0.00428	1
mlle	10	10	0.00428	1
srp30+ltsa	10	70	0.00432	1
ltsa	15	70	0.00432	1
srp30+ltsa	25	50	0.00424	1
srp30+hlle	10	70	0.00428	1
ltsa	25	50	0.00428	1
ltsa	40	50	0.00424	1
srp30+ltsa	15	50	0.00424	1
ltsa	10	70	0.00432	1
dummy			0.00408	113.74

4.5. Rasprava

U prošlom odjeljku težina je bila na specifičnim skupovima podataka, dok je u ovom odjeljku cilj iznijeti generalnije zaključke. Ono što je proizašlo iz rezultata je da promjena parametara i metoda korištenih drastično utječe na rezultat. Ovo odgovara poznatom teoremu o nepostojanju besplatnog ručka (engl. *"No Free Lunch" theorem*) koji kaže da ne postoji jedan algoritam koji ispunjava potrebe svih problema.

Također, pokazalo se kako smanjivanje dimenzije metodom rijetkih slučajnih projekcija prije nego što se izvodi učenje mnogostrukosti, osim što smanjuje trajanje izvođenja algoritma, dodatno i povećava performanse u kontekstu analize koja koristi sporedna sredstva. Postoji manjak empirijskih istraživanja na ovu temu te ona definitivno prezentira razne mogućnosti.

Za korištene skupove podataka pokazalo se kako, iako model HW ima značajno veću točnost, koristeći mjeru entropije pogađanja, u ovom kontekstu boljim se modelom pokazao model value. Kod modela HW je za očekivati veća točnost klasifikacije jer je jednostavnije napraviti klasifikaciju u 9 razreda nego u 256 razreda. Što se tiče boljih ukupnih performansi na value modelu, jedno moguće objašnjenje je da ako na kreaciju sporednih sredstava gledamo kao na transformaciju originalnih podataka P u sporedna sredstva S, $\Phi : P \mapsto S$, na učenje mnogostrukosti možemo gledati kao traženje inverzne transformacije $\Phi^{-1} : S \mapsto P$. Ovaj zadatak je lakši kada imamo direktne vrijednosti dobivene mjerenjem nego kada je uvedena dodatna razina, u ovom slučaju Hammingova težina. Bolju efikasnost napada moguće je postići optimizacijom hiperparametara klasifikatora ili korištenjem dodatnih klasifikatora.

5. Zaključak

U ovom diplomskom radu razmatrala se primjena postupaka učenja mnogostrukosti za povećavanje učinkovitosti analize koja koristi sporedna sredstva kriptografskih uređaja (SCA) te su detaljno opisana polja analize sporednih sredstava i smanjenja dimenzionalnosti. Za SCA dan je uvid u ovo područje, opisane su osnove sporednih sredstava i napravljena je taksonomska podjela područja. Nakon toga su opisani razni napadi te širina njihove primjene. Zatim su opisane i razne implementirane protumjere protiv napada SCA. Uz SCA proučena je i ideja smanjenja dimenzionalnosti koja je u ovom radu podijeljena na dvije velike podskupine, a to su linearno smanjenje dimenzionalnosti i učenje mnogostrukosti (nelinearno smanjenje dimenzionalnosti). Naglasak u ovom radu je na metodama učenja mnogostrukosti te je iz skupine linearnog smanjenja kao jedna od referentnih metoda uzeta i vrlo popularna metoda PCA. Osim PCA, dodatna referentna vrijednost u ovom radu je kako se izvedeni eksperimenti ponašaju kada nije implementirano nikakvo smanjenje dimenzionalnosti. Uz ovo, korištena je dodatno hibridna metoda smanjenja dimenzionalnosti koja prije metoda učenja mnogostrukosti izvršava jednostavniju metodu rijetkih slučajnih projekcija kako bi se napravila transformacija u međuprostor na kojem se onda provode opisane metode. Iz dobivenih rezultata vidljivo je da su korištene metode vrlo osjetljive na promjenu hiperparametara te je njihova optimizacija ključna za postizanje dobrih rezultata. Također, generalno se pokazalo kako je bolje koristiti model koji uzima međuvrijednosti u ovom kontekstu, nego model s Hammingovom težinom, koji se češće smatra boljim. Prezentirana hibridna metoda pokazala se kao izvrstan način za ne samo skraćivanje vremena izvođenja, već i za postizanje boljih rezultata. Dodatne kombinacije jednostavnijih i složenijih metoda su moguće koje imaju potencijal rezultirati i boljim performansama. Primjerice, umjesto rijetkih slučajnih projekcija u istu se svrhu mogu koristiti i gaussianske slučajne projekcije. Korištenje ovakvih metoda moglo bi rezultirati još boljim rezultatima na višedimenzionalnim skupovima i na njima uvesti velike doprinose u kontekstu efikasnosti algoritma za smanjenje dimenzionalnosti. Također, bolji rezultati napada mogu se postići odabirom specifičnog klasifikatora i njegovih pa-

rametara za željenu metodu smanjenja dimenzionalnosti. Ovo nije bilo u okviru teme ovog rada, jer je cilj bio usporediti razne tehnike učenja mnogostrukosti u kontekstu SCA. Usprkos tome, na svim skupovima uspješno je postignuta dovoljno niska entropija pogađanja koristeći relativno mali originalni skup značajki veličine 50. Na skupu podataka koji implementira slučajno kašnjenje kao protumjeru postignuti su odlični rezultati te je dovoljno niska vrijednost entropije pogađanja postignuta nakon prosječno 2000 tragova.

LITERATURA

Ross Anderson, Mike Bond, Jolyon Clulow, Sergei Skorobogatov. Cryptographic processors-a survey. *Proceedings of the IEEE*, 94(2):357–369, 2006.

Mukund Balasubramanian, Eric L Schwartz, Joshua B Tenenbaum, Vin de Silva, John C Langford. The isomap algorithm and topological stability. *Science*, 295 (5552):7–7, 2002.

Mikhail Belkin Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the rsa encryption standard pkcs# 1. U *Annual International Cryptology Conference*, stranice 1–12. Springer, 1998.

Gerrit Bleumer. *Blinding Techniques*, stranice 150–152. Springer US, Boston, MA, 2011. ISBN 978-1-4419-5906-5. doi: 10.1007/978-1-4419-5906-5_182. URL `https://doi.org/10.1007/978-1-4419-5906-5_182`.

Bo Yang, Kaijie Wu, Ramesh Karri. Scan based side channel attack on dedicated hardware implementations of data encryption standard. U *2004 International Conference on Test*, stranice 339–344, 2004.

Dan Boneh, Richard A DeMillo, Richard J Lipton. On the importance of checking cryptographic protocols for faults. U *International conference on the theory and applications of cryptographic techniques*, stranice 37–51. Springer, 1997.

Ingwer Borg Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

Eric Brier, Christophe Clavier, Francis Olivier. Correlation power analysis with a leakage model. U *International workshop on cryptographic hardware and embedded systems*, stranice 16–29. Springer, 2004.

David Brumley Dan Boneh. Remote timing attacks are practical. *Computer Networks*, 48(5):701–716, 2005.

John P Cunningham Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research*, 16(1): 2859–2900, 2015.

Jean-Francois Dhem, Francois Koeune, Philippe-Alexandre Leroux, Patrick Mestré, Jean-Jacques Quisquater, Jean-Louis Willems. A practical implementation of the timing attack. U *International Conference on Smart Card Research and Advanced Applications*, stranice 167–182. Springer, 1998.

David L. Donoho Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003. ISSN 0027-8424. doi: 10.1073/pnas.1031596100. URL `https://www.pnas.org/content/100/10/5591`.

Gordon Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE transactions on information theory*, 14(1):55–63, 1968.

John Kelsey, Bruce Schneier, David Wagner, Chris Hall. Side channel cryptanalysis of product ciphers. U *European Symposium on Research in Computer Security*, stranice 97–110. Springer, 1998.

P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, Y. Yarom. Spectre attacks: Exploiting speculative execution. U *2019 IEEE Symposium on Security and Privacy (SP)*, stranice 1–19, 2019.

Paul Kocher, Joshua Jaffe, Benjamin Jun. Differential power analysis. U *Annual International Cryptology Conference*, stranice 388–397. Springer, 1999.

Paul Kocher, Ruby Lee, Gary McGraw, Anand Raghunathan. Security as a new dimension in embedded system design. U *Proceedings of the 41st annual Design Automation Conference*, stranice 753–760, 2004.

Paul Kocher, Joshua Jaffe, Benjamin Jun, Pankaj Rohatgi. Introduction to differential power analysis. *Journal of Cryptographic Engineering*, 1(1):5–27, 2011.

Paul C Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. U *Annual International Cryptology Conference*, stranice 104–113. Springer, 1996.

Boris Köpf David Basin. An information-theoretic model for adaptive side-channel attacks. U *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, stranica 286–296, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937032. doi: 10.1145/1315245.1315282. URL `https://doi.org/10.1145/1315245.1315282`.

Markus G Kuhn. Optical time-domain eavesdropping risks of crt displays. U *Proceedings 2002 IEEE Symposium on Security and Privacy*, stranice 3–18. IEEE, 2002.

Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, Mike Hamburg. Meltdown: Reading kernel memory from user space. U *27th USENIX Security Symposium (USENIX Security 18)*, 2018.

Joe Loughry David A Umphress. Information leakage from optical emanations. *ACM Transactions on Information and System Security (TISSEC)*, 5(3):262–289, 2002.

Laurens van der Maaten Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

J. L. Massey. Guessing and entropy. U *Proceedings of 1994 IEEE International Symposium on Information Theory*, stranice 204–, 1994.

JM McConnell. National security telecommunications and information systems security. *Retrieved May*, 6:2007, 1992.

Leland McInnes John Healy. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv*, abs/1802.03426, 2018.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2 (11):559–572, 1901.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

S. Picek, A. Heuser, A. Jovic, L. Batina. A systematic evaluation of profiling through focused feature selection. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(12):2802–2815, 2019.

Todd Rowland. Manifold. URL `https://mathworld.wolfram.com/Manifold.html`.

Lawrence K Saul Sam T Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of machine learning research*, 4(Jun):119–155, 2003.

Semiconductor Engineering. Scan test, Dec 2019. URL `https://semiengineering.com/knowledge_centers/test/scan-test-2/`.

Adi Shamir Eran Tromer. Acoustic cryptanalysis. *presentation available from http://www. wisdom. weizmann. ac. il/ tromer*, 2004.

François-Xavier Standaert. Introduction to side-channel attacks. U *Secure Integrated Circuits and Systems*, stranice 27–42. Springer, 2010.

François-Xavier Standaert, Tal G Malkin, Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. U *Annual international conference on the theory and applications of cryptographic techniques*, stranice 443–461. Springer, 2009.

Chin Chi Tiu. *A new frequency-based side channel attack for embedded systems*. Doktorska disertacija, University of Waterloo, 2005.

Wikimedia Commons. Tangentialvektor.svg, 2008. URL `https://commons.wikimedia.org/wiki/File:Tangentialvektor.svg`. [Dohvaćeno 7. lipnja 2020.].

S. Zhang, Z. Ma, H. Tan. On the equivalence of hlle and ltsa. *IEEE Transactions on Cybernetics*, 48(2):742–753, 2018.

Zhenyue Zhang Jing Wang. Mlle: Modified locally linear embedding using multiple weights. U *Advances in neural information processing systems*, stranice 1593–1600, 2007.

YongBin Zhou DengGuo Feng. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. *IACR Cryptology ePrint Archive*, 2005(388), 2005.

Tehnike učenja višestrukosti za povećanje učinkovitosti analize koja koristi sporedna svojstva kriptografskih uređaja

Sažetak

Cilj ovog diplomskog rada je razmotriti primjenu postupaka učenja mnogostrukosti kako bi se povećala učinkovitost analize koja koristi sporedna sredstva kriptografskih uređaja (SCA). U sklopu toga detaljno su opisana polja SCA i smanjenja dimenzionalnosti te za njih specifične metode korištene u tom polju. Iako je naglasak u ovom radu na metodama učenja mnogostrukosti, kao jedna od referentnih metoda uzeta je i vrlo popularna metoda za smanjenje dimenzionalnosti, PCA. Dodatno je korištena hibridna metoda smanjenja dimenzionalnosti koja prije metoda učenja mnogostrukosti izvodi metodu rijetkih slučajnih projekcija kako bi se napravila transformacija u međuprostor. Za konačnu klasifikaciju koristi se metoda slučajnih šuma zbog velike otpornosti na smetnje.

Ključne riječi: učenje mnogostrukosti, analiza koja koristi sporedna svojstva (SCA), entropija pogađanja (GE), smanjenje dimenzionalnosti, AES

Manifold Learning Techniques for Increased Efficiency of Side-Channel Analysis

Abstract

This master's thesis aims to study the application of manifold learning techniques with the intent of increasing the efficiency of side-channel analysis (SCA). With this in mind, both SCA and dimensionality reduction fields, along with their respected practices, are described. Even though the emphasis in this paper is on manifold learning, as one of the reference methods, the highly popular PCA method for dimensionality reduction was also considered. Additionally, a hybrid dimensionality reduction method was applied, in which, before the manifold learning method, sparse random projections are performed to transform data into interspace. The random forest method is used for the final classification due to its high resistance to noise.

Keywords: manifold learning, side-channel analysis (SCA), guessing entropy (GE), dimensionality reduction, AES