hyväksymispäivä arvosana

arvostelija

Roskapostin torjunta- ja luokittelumenetelmät

Jukka Huhta

Helsinki 3.10.2011

Pro gradu -tutkielma

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen	Tietojenkäsittelytieteen laitos

Tekijä — Författare — Author
Jukka Huhta

Työn nimi — Arbetets titel — Title
Roskapostin torjunta- ja luokittelumenetelmät

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Tutkielma	3.10.2011	66 sivua

Tiivistelmä — Referat — Abstract

Tässä tutkielmassa tutustutaan kirjallisuuden avulla yleisesti käytössä oleviin roskapostin torjuntamenetelmiin. Myös niitä soveltava järjestelmäkokonaisuus esitellään. Työssä käsitellään esimerkiksi mustat DNS-listat, kollaboratiivisia tekniikoita ja harmaalistaus. Sisältöpohjaisiin menetelmiin, erityisesti bayesiläiseen luokitteluun ja logistiseen regressioanalyysiin tutustutaan tarkemmin. Tutkielmassa perehdytään myös roskapostitusta rajoittavaan lainsäädäntöön ja pohditaan, minkälaisilla keinoilla päädyttäisiin kokonaisuuden kannalta parhaaseen lopputulokseen. Työn kokeellisessa osuudessa verrataan logistista regressioanalyysiä ja bayesiläistä luokittelua roskapostintunnistuksessa realistisella koeasetelmalla käyttäen aitoa sähköpostikorpusta aineistona. Tärkeimmät kokeisiin perustuvat johtopäätökset ovat, että logistiseen regressioanalyysiin pohjaava tunnistus täydentäisi luokittelutuloksen puolesta erinomaisesti roskapostintorjuntajärjestelmää bayesiläisen luokittelijan rinnalla, mutta menetelmänä se on liian hidas tietokantanoudoista johtuvan I/O-vaativuuden takia. Lisäksi todetaan, että jopa käytettyä luokittelumenetelmää tärkeämpi seikka oppivaa roskapostintunnistusta hyödyntävässä järjestelmässä saattaa olla luokittelijalle syötetty aineisto, jonka laadun varmistamiseen on syytä panostaa erityisesti monen käyttäjän roskapostintorjuntajärjestelmässä, jossa luokitellaan kaikkien käyttäjien viestit samaan aineistoon perustuen.

ACM Computing Classification System (CCS):
C.4 [Performance of systems],
G.3 [Probability and statistics],
I.5.4 [Pattern recognition: Applications],
K.4.1 [Computers and society: Public Policy Issues],

Avainsanat — Nyckelord — Keywords
roskaposti, bayesiläinen luokittelu, logistinen regressioanalyysi, estolistat, SpamAssassin

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Sisältö

1 Johdanto

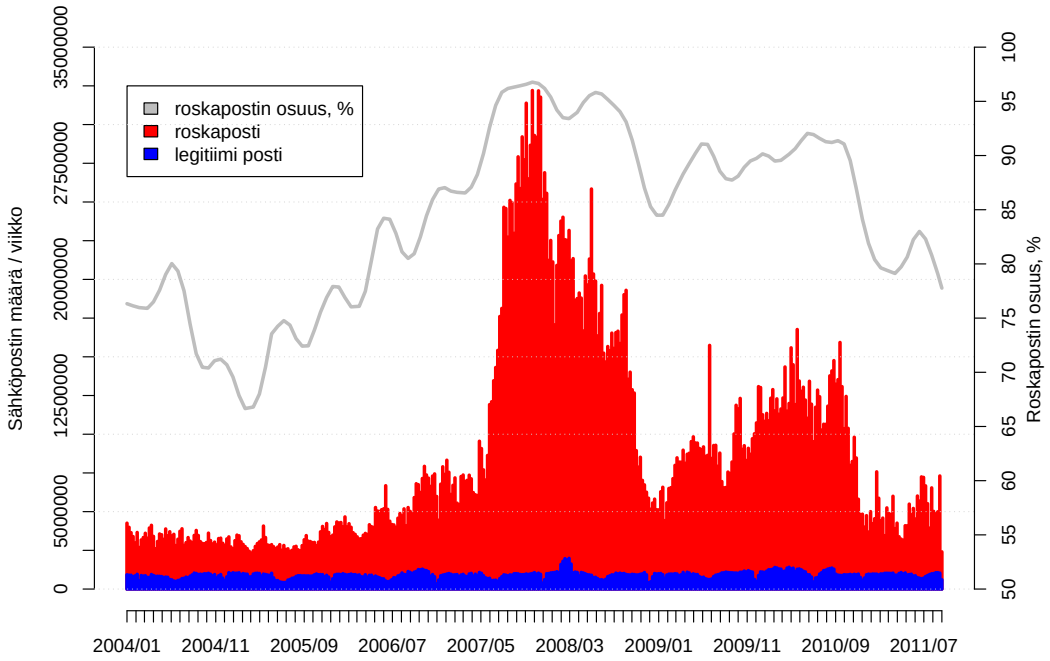
Sähköpostista on muodostunut viime vuosikymmenten aikana erittäin merkittävä viestintäväline yrityksissä, oppilaitoksissa, muutenkin julkisella sektorilla ja yksityisviestinnässä. 1990-luvun loppupuolelta lähtien roskapostista on kuitenkin tullut yhä enemmän sähköpostin asianmukaista käyttöä haittaava tekijä. Se on lisännyt organisaatioiden kustannuksia ylläpito- ja laitteistoresursseina sekä sähköpostin käyttäjien hukkaaman ajan muodossa.

Roskapostin – eli ilman vastaanottajan suostumusta massapostitetun sähköpostin – määrä on myös lisääntynyt kuluneen vuosikymmenen aikana voimakkaasti. Erään tutkimuksen mukaan tammikuussa 2003 kaikesta sähköpostista 24 prosenttia oli roskapostia, kun maaliskussa 2005 luku oli jo 83 prosenttia (Gomes et al., 2007). Kasvu ei kuitenkaan ole tasaista eikä edes jatkuvaa. Aivan viime vuosina roskapostin määrä näyttäisi sijoittuneen noin 80–90 prosenttiin kokonaisvolyymistä, vaikka tilanne vaihtelee voimakkaasti organisaatiosta toiseen. Esimerkiksi Helsingin yliopiston tietojenkäsittelytieteen laitos on ilmoittanut[1], että yli 95 prosenttia sen postijärjestelmään saapuvasta postista on roskapostia. Koko yliopiston[2] osalta tilanne hahmottuu kuvasta 1. Roskapostin suhteellinen osuus on pitkään ollut yli 90 prosenttia, mutta vuoden 2011 alusta selvästi laskussa. Laajemmin tilannetta seuraava Messaging Anti-abuse Working Group arvioi loppuvuoden 2010 raportissaan (MAAWG, 2011) luvun olevan noin 90 % tai hieman alle.

Vain hieman kärjistäen voidaan siis todeta, että sähköpostijärjestelmien käsittelykapasiteetista suurin osa on varattu roskapostien käsittelylle, mistä aiheutuu suoraan merkittäviä laitteisto- ja henkilöstökuluja. Postia välittävät ja vastaanottavat laitteistot on mitoitettava vieläpä ruuhkahuippujen mukaan, jotta ongelmatilanteissakaan – kuten hajautettua palvelunestohyökkäystä muistuttavassa roskapostinlähetyspurskeessa – legitiimin sähköpostin toimittaminen ei kohtuuttomasti viivästyisi. Mitä monimutkaisempia viestien sisältöön perustuvia ja esimerkiksi koneoppimista hyödyntäviä luokittelumenetelmiä käytetään, sitä enemmän laskentaresursseja roskapostien suodattaminen vaatii. Suuren postivolyymin järjestelmät tuleekin suunnitella siten, että haittaposteista mahdollisimman suuri osa torjutaan mahdollisimman varhain ja kevyin menetelmin. Tällainen menetelmä on nimipalvelutekniikkaa hyödyntävät mustat listat (DNSBL), joiden avulla voidaan estää suuri osa roskapostista lähettäjän IP-osoitteen perusteella, ottamatta kantaa viestien

[1] `http://www.cs.helsinki.fi/compfac/roskaposti.html`, noudettu 3.5.2011.

[2] Kirjoittaja toimii työkseen Helsingin yliopiston sähköpostijärjestelmien ylläpidossa.



Kuva 1: Roskapostiksi tunnistetun tai sellaisena torjutun ja legitiimin postin määrä sekä roskapostin suhteellinen osuus tammikuusta 2004 syyskuuhun 2011 Helsingin yliopiston postijärjestelmässä.

sisältöön.

Laitteistojen mitoituksen lisäksi roskapostin aiheuttamiin kustannuksiin on laskettava esimerkiksi sähköpostinkäyttäjien hukkaaman ajan määrä, virheellisten tunnistusten ja läpi päästettyjen roskapostien aiheuttamat aiheellisten postien katoamiset, sähköpostihuijauksiin langenneiden käyttäjien aikaansaamien tietoturvapoikkeamien ja jopa suorien rahanmenetysten selvittely ja kaikkiin näihin liittyvä tietotekniikkatukiorganisaatioiden ylimääräinen kuorma.

Näitä kaikkia haittoja ja kustannuksia punnitessa näyttää ilmeiseltä, että roskapostit on torjuttava mahdollisimman tehokkaasti ja mahdollisimman varhain. Ongelmia ja reunaehtoja roskapostin tehokkaalle torjumiselle aiheuttavat roskapostintunnistuksen teknisen haastavuuden ja tietojenkäsittelykapasiteetin riittävyyden lisäksi esimerkiksi ylläpitoresurssien mitoitus sekä lainsäädännön ja muiden säädös-

ten takaama sähköpostin käyttäjien yksityisyyden suoja. Kuitenkin on pyrittävä mahdollisimman tarkkaan lopputulokseen eli roskapostintunnistukseen, joka tekee enintään siedettävän määrän virheitä. Sopivaa virhekynnystä on tosin vaikeaa ennalta määritellä, koska virheellisen tunnistuksen aiheuttama haitta on niin ikään vaikea määritellä formaalisti. Lisäksi luokittelujen oikeellisuutta on jälkikäteenkin lähes yhtä vaikeaa arvioida koneellisesti kuin reaaliajassa eli roskapostin suodatuksen yhteydessä. Käytännössä usein päädytäänkin muuttamaan järjestelmien parametrejä toivottuun suuntaan käyttäjiltä saadun palautteen ja muiden kokemusten perusteella. Käyttäjäpalautteen eli ainakin virheellisten tunnistusten systemaattinen kerääminen onkin oppivien luokittelumenetelmien pitkäaikaisen käytön edellytys.

Tässä tutkielmassa käydään läpi yleisesti käytettyjä roskapostintorjuntamenetelmiä. Perinteisesti viestien sisältöön perustuvassa roskapostien luokittelussa on käytetty niin sanottua bayesiläistä menetelmää, joka tosin on muodostunut lähes yleiskieliseksi nimitykseksi kaikelle oppivalle roskapostintorjunnalle, vaikka todellisuudessa käytössä on paljon muitakin algoritmeja. Tuoreessa tutkimuskirjallisuudessa mielenkiintoa on herättänyt esimerkiksi tukivektorikoneiden soveltaminen sähköpostien luokitteluongelmaan, mutta niitä on myös kritisoitu liian raskaiksi ja siksi huonosti soveltuviksi käytännön roskapostintorjuntajärjestelmiin. Perehdymme lähemmin myös kirjallisuudessa esitettyyn logistisen regressioanalyysin menetelmään ja testaamme kokeellisesti, kuinka hyvä tarkkuus ja suorituskyky sillä olisi käytännössä ja erityisesti yleisesti käytetyn SpamAssassin-torjuntaohjelmiston osana ja verrattuna sen bayesiläiseen luokittelutoteutukseen.

Testituloksista havaitsemme, että menetelmä on tunnistustarkkuuden osalta hyvä ja täydentäisi keinovalikoimaa hyvin, jopa ajettuna bayesiläisen luokittelun rinnalla samalla aineistolla opetettuna, mutta logistisen regressioanalyysin toteuttavan komponentin suoritus on niin hidasta, ettei sitä voi käytännössä suositella suuremmassa tuotantoympäristössä. Oleellisempaa luokittelutarkkuuden lisäämiseksi lieneekin esimerkiksi opetusmateriaalin laadun parantaminen.

Työ on järjestetty seuraavasti. Luvussa 2 tutustutaan roskaposti-ilmiöön, kuten sen syihin ja seurauksiin ja torjuntaan. Luvussa 3 esitellään käytettyjä ja hyväksi havaittuja roskapostin torjuntamenetelmiä. Luvussa 4 tutustutaan syvemmin oppiviin tunnistusmenetelmiin. Luvussa 5 syvennytään näiden menetelmien ja roskapostintorjunnan ongelmakohtiin ja luvussa 6 testataan kokeellisesti, miltä osin logistinen regressio voisi menetelmänä vastata eräisiin näistä ongelmakohdista. Lopuksi kerrataan johtopäätöksiä.

2 Roskaposti ilmiönä

Roskaposti Internetissä on lähes yhtä vanha ilmiö kuin SMTP-protokollaan (Postel, 1982) ja uutisryhmiin (Usenet News) perustuva sähköinen viestintä. Itse asiassa vanhempikin; monien lähteiden (esim. Zdziarski, 2005) mukaan ensimmäinen roskaposti – The DEC spam – lähetettiin vuonna 1978 Arpanetissä, joka on Internetin edeltäjä. Varhaisina vuosina roskaposteja lähetettiin kuitenkin harvakseltaan ja viestit olivat yksittäistapauksia, joten erityisiä tietojenkäsittelyllisiä ratkaisujakaan ei tarvittu.

Vasta 1990-luvun puolivälin jälkeen, erityisesti vuosina 1996 ja 1997, roskapostiongelma alkoi näkyä kunnolla, kuten aikalaisraporteista (esim. Junod, 1997) voi päätellä. Tällöin vielä postipalvelimet olivat usein avoimia välityspalvelimia, jolloin postia saattoi kuka tahansa lähettää käyttäen kenen tahansa palvelinta ja esiintyen kenenä tahansa. Tämä ongelma korjaantui sulkemalla postinvälitys asiattomilta tahoilta; keino, jota yllä mainitussa Junodin artikkelissakin esitetään.

Varsinaisina suodatuskeinoina käytettiin tuolloin usein käyttäjäkohtaisia avainsanalistoja, ja mikäli esimerkiksi viestin otsikosta tai sisällöstä löytyi määritelty sana tai merkkijono, viesti suodatettiin roskapostina. Esimerkiksi yliopistoissa, joiden UNIX-järjestelmissä käyttäjät pääsivät määrittelemään Procmaililla omia, usein säännöllisiin lausekkeisiin perustuvia suodatussääntöjään, tämä oli suosittua.

Vuonna 1998 perustettiin nimipalveluun perustuvia mustia (DNSBL) listoja tarjoava Spamhaus-projekti[3], joka oman ilmoituksensa mukaan suojaa peräti 1.4 miljardia postilaatikkoa. Myös Helsingin yliopisto käyttää kyseisen organisaation tarjoamia mustia listoja.

Vasta vuosituhannen vaihteen tienoilla alettiin ehdottaa erilaisia oppivia menetelmiä roskapostin tunnistamiseksi, erityisesti naiivia bayesiläistä menetelmää (Androutsopoulos et al., 2000a), johon palaamme jäljempänä. Paul Grahamin kuuluisan esseen A Plan for Spam (2002) jälkeen bayesiläistä menetelmää alettiin hyödyntää monissa yleisesti käytetyissä roskapostinsuodatus- tai tunnistusohjelmissa, kuten SpamAssassin ja Bogofilter.

Tässä työssä keskitytään Internetin sähköpostin yhteydessä esiintyvään roskapostiin, *e-mail spam*, vaikka sama asiattomasti massalahetettyjen viestien ongelma koskettaa myös blogeja, wikejä, keskustelupalstoja ja monia muita www-pohjaisia verkkopalveluita. Niiden ehkäisy- ja torjuntamenetelmien valikoima kuitenkin poikkeaa jonkin verran perinteisemmästä roskapostista, vaikka yhtymäkohtiakin on.

[3] http://www.spamhaus.org/organization/

Roskaposti-ilmiön ymmärtämiseksi on syytä todeta muutama sana roskapostin luonteesta. Ensinnäkin, kyse on nykyään tyypillisimmin rahallisen hyödyn tavoittelusta, joka on helppoa, koska toiminnan enimmät kustannukset on ulkoistettu vastaanottajille. Tähän palaamme tarkemmin aliluvussa 2.2.

Toiseksi, roskapostittajien ja roskapostin torjujien välillä vallitsee jatkuva varustelukilpa. Roskapostittajat pyrkivät vastaamaan kaikkiin suodatustoimenpiteisiin jollakin konstilla, jonka tarkoitus on kiertää suodatus. Muun muassa tähän perustuu yleinen käsitys, jonka mukaan roskaposti muuttaa muotoaan jatkuvasti. Sullivan (2004) esitti, että roskapostisukupolven pituus olisi noin kolme kuukautta. Hän jäsensi roskapostiaineistonsa joukoksi erilaisia ominaisuuksia, ja seurasi, kuinka tämä joukko uudistuu. Calais Guerra et al. (2008) vahvistivat saman tutkittuaan SpamAssassin-torjuntaohjelman eri versioita ja eri-ikäisiä roskapostiaineistoja.

Kolmanneksi, roskapostitus on usein kampanjaluontoista. Erityisesti kalastelukampanjat kestävät vain lyhyen aikaa, jopa alle tunteja. Muut roskapostituskampanjat voivat kestää pidempäänkin, mutta yleensä rajallisen ajan (esim. Zheleva et al., 2008; Sheng et al., 2009; Taylor et al., 2007). Tämä myös tarkoittaa, että roskapostin määrä näkyy yleensä vastaanottavilla palvelimilla purskeina.

Viime vuosina nähtyä roskapostituksen kehitystä luonnehtivat ainakin seuraavat seikat:

- Roskapostin lähteinä toimivat suurelta osalta ns. botnetit (Bradbury, 2006).
- Kuvaroskaposti on vähentynyt käytännössä olemattomiin.
- Myös suomenkielisiä roskaposteja on alkanut esiintyä, mutta lähinnä konekäännöksiä. Ilmiö alkoi tulla näkyviin alan yritysten seurantaraporteissa viime vuosikymmenen lopulla (McAfee, 2008; MessageLabs Intelligence, 2009).
- Roskapostin määrä ylipäänsä on aivan viime aikoina vähentynyt radikaalisti (ks. kuva 1). Syynä on ilmeisesti eräiden suurten toimijoiden kiinnijäänti.

Seuraavissa aliluvuissa määritellään tässä työssä – myös yllä – käytettäviä käsitteitä ja termejä, perehdytään roskaposti-ilmiön taustalla oleviin syihin ja seurauksiin sekä roskapostitusta ja sen suodatusta koskevaan lainsäädäntöön. Pintapuolisesti esitellään myös roskapostin torjunnassa ja suodatuksessa käytettäviä tekniikoita, joihin palataan tarkemmin myöhemmin.

2.1 Määritelmiä ja termejä

Roskaposti (engl. spam, junk e-mail) määritellään tavallisesti esimerkiksi pyytämättä massalähetetyiksi sähköposteiksi (engl. Unsolicited Bulk Email, UBE). Määritelmiä on paljon muitakin, ja eräs täsmällinen määritelmä kuuluu näin:

> Pyytämätön, ei-toivuttu sähköposti, joka on lähetetty valikoimattomasti, suoraan tai kiertoteitse, ilman että lähettäjällä on senhetkistä suhdetta vastaanottajaan. (Cormack ja Lynam, 2005)

Yhtä hyvin roskaposti voidaan laveasti määritellä informaatioksi, joka ei tarjoa lisäarvoa käyttäjälle. Tarkemmin sanottuna sellaista voisi olla sopimaton, pyytämätön, toistuva ja epäoleellinen sisältö (Hayati ja Potdar, 2008). Tämänkaltaista määritelmää voi soveltaa myös muualla Internetissä esiintyvään roskapostiin.

Myös puutteellisia määritelmiä käytetään. Usein puhutaan pyytämättömästä kaupallisesta sähköpostista (engl. Unsolicited Commercial Email, UCE), mutta tätä määrittelyä pidetään yleisesti riittämättömänä ja jopa virheellisenä ainakin kolmesta syystä:

1. Myös epäkaupallinen viestintä, kuten hyväntekeväisyyteen, koulutukseen, uskontoon tai politiikkaan liittyvät massapostitukset, ovat tavallisia.

2. Roskapostia arvioitaessa ei tule ottaa kantaa sisältöön vaan vain suostumukseen – näin vältetään myös monet sananvapauteen liittyvät ongelmat. ”Spam is about consent, not content”, todetaan usein keskusteltaessa roskapostin määritelmästä (esim. Spamhaus Project, 2003).

3. Monet pyytämättömät viestit, kuten ensikontaktit, työtarjoukset tai tiedustelut eivät ole roskapostia. Roskapostin täytyy olla myös massapostitettua.

UCE-määritelmään kuitenkin perustuu osa roskapostitusta rajoittavasta lainsäädännöstä, kuten seuraavassa aliluvussa havaitsemme. Siitä seuraa ongelmia ainakin lainsäädännön kattavuudelle.

Yllä mainitut määritelmät eivät muunnu ongelmitta koneluettavaan muotoon. Arkikokemuksesta nimittäin tiedämme, että jokin viesti voi olla yhdelle roskaposti ja toiselle ei – riippuen esimerkiksi juuri siitä, onko suostumusta annettu vai ei. Vastaanottajan suostumus tai sen puuttuminen on oleellisin yksittäinen tieto, jonka perusteella päätetään, onko viesti roskapostia. Suostumus ei käy ilmi itse viestistä,

tai jos käy, tietoon ei voi luottaa. Koneellisten tunnistusmenetelmien on siis tehtävä päätös muilla perusteilla.

Usein koneellisen roskapostintunnistuksen tekemä päätös on väärä, eli asiallinen viesti, joka ei ole roskapostia, tulkitaan roskapostiksi. Tällaista tunnistusta kutsutaan *vääräksi positiiviseksi*, FP (engl. false positive). Vastaavasti roskaposti, jota ei tunnistettu sellaiseksi, on *väärä negatiivinen*, FN (engl. false negative). Oikein tunnistunut roskaposti on oikea positiivinen, TP ja ei-roskaposti oikea negatiivinen, TN (engl. true positive/negative). Tavallisia sähköpostiviestejä, jotka eivät ole roskaposteja, kutsutaan toisinaan myös *legitiimeiksi viesteiksi* (engl. non-spam, ham). Vääriä positiivisia pidetään yleensä paljon vääriä negatiivisia pahempana virheenä (kuten Androutsopoulos et al., 2000a), joskin ero on subjektiivinen ja siksi vaikea mitata.

Roskapostiviestit voidaan jakaa useampiin alalajeihin, esimerkiksi tarkoituksen tai huijaustyypin mukaan. *Perinteisillä roskaposteilla* yritetään myydä jotain, kuten kopiotuotteita, ohjelmistoja, lääkkeitä, tutkintotodistuksia tai pornoa. *Virus- ja haittaohjelmaviestit* ovat oma lajinsa, jota haitallisuutensa vuoksi torjutaan roskapostisuodatuksen lisäksi erillisillä torjuntaohjelmilla. Oma lajityyppinsä ovat ns. *nigerialaiskirjeet* (engl. 419 scams), joissa pyydetään yleensä siirtämään joku summa rahaa isomman aarteen toivossa. Lottohuijaukset ovat näille läheistä sukua. Nettihuutokauppojen, pankkien, webmail-palvelujen ja muiden vastaavien palveluja tunnuksia urkitaan *kalasteluviesteillä* (engl. phishing scam messages), jotka ovat melko yleisiä ja samalla myös niin haitallinen roskapostin alalaji, että monet organisaatiot kiinnittävät niihin erityistä huomiota suodatustoimissaan (esim. Taylor et al., 2007). Paljon muitakin roskapostityyppejä on, kuten poliittinen tai uskonnollinen propaganda, ketjukirjeet sekä erilaiset ilkivaltaisesti lähetetyt viestit, joiden tarkoitus on esimerkiksi lamaannuttaa oppivat suodattimet (engl. bayes poisoning) tai tukkia vastustajan laatikko virheilmoituksin väärentämällä tämän osoite viestien lähettäjäksi.

Erillisenä roskapostilajina mainittakoon *kuvaroskaposti*, joka voi sisällöllisesti olla mitä tahansa roskapostia; oleellista on erikoinen muoto, jolla pyritään hämäämään suodattimia. Viesteissä varsinainen tekstiosuus on olematon tai sisältää täysin asiaan liittymätöntä tekstiä, jotteivät viestin sisältöön huomiota kiinnittävät suodattimet reagoisi. Viestin varsinainen hyötykuorma on upotettu liitetiedostona välitettyyn kuvaan, joka on vieläpä laadittu CAPTCHA:n tapaan (Completely Automated Public Turing test to tell Computers and Humans Apart), jottei optinen teks-

tintunnistus onnistuisi. Tällaiset kuvaroskapostit muodostivat pahimmillaan jopa puolet lähetetyistä roskaposteista (Conley, 2007). Sittemmin ilmiö näyttäisi lähes kadonneen, mutta mikään ei estä sitä toistumasta.

Roskaposteista suuri osa lähetetään käyttäen niin sanottuja *botnetteja*. Botnet koostuu joukosta murrettuja tietokoneita, *zombeja*, joita ohjaillaan käyttämällä IRCiä komentokanavana. Murretut koneet ovat tyypillisesti Windows-käyttöjärjestelmällä varustettuja, laajakaistayhteyden päässä olevia kotikoneita, joiden tietoturvasta ei ole huolehdittu asianmukaisesti (Puri, 2003; Rajab et al., 2006). Botnetteja käytetään monenlaisiin tarkoituksiin, kuten tietomurtoihin, hajautettuihin palvelunestohyökkäyksiin (DDoS), virusten levittämiseen ja ennen muuta roskapostitukseen.

2.2 Roskapostituksen syyt ja seuraukset

Roskapostitus on erittäin yleinen ilmiö yksinkertaisesti siksi, että joku aina hyötyy siitä taloudellisesti. Ansaintalogiikka vaihtelee samoin kuin roskapostituksen tarkoitus sekä viestien rakenne ja sisältö, mutta joka tapauksessa roskapostitus näyttäisi kytkeytyvän monin tavoin muuhun nettirikollisuuteen. Näin on toisaalta siksi, että samat keinot soveltuvat muuhunkin nettirikollisuuteen: koneiden kaappaaminen ja siten hankitut botnetit soveltuvat yhtä lailla roskapostitukseen kuin virusten levitykseen, hajautettuihin palvelunestohyökkäyksiin ja tietomurtoihin. Toisaalta, aivan kuten monet muutkin yhteiskunnalliset ilmiöt, myös osa rikollisuudesta on siirtynyt viime aikoina hyödyntämään Internetiä. Niinpä esimerkiksi petoksia yritetään tehtailla roskapostittamalla tunteisiin vetoavia viestejä.

Rikos kannattaa, sillä roskapostitus ei juurikaan maksa roskapostittajalle. Vaikka roskaposteja suodatettaisiin, uusien viestien lähettämisen rajakustannus on nolla tai lähes nolla (Klensin, 2005). Miljoonien sähköpostiosoitteiden listoja voi ostaa muutamilla dollareilla, eikä roskapostittamiseen soveltuvan botnetin vuokraaminen maksa juuri enempää. Kustannukset koituvat lähinnä vastaanottajien maksettavaksi, suoraan tai epäsuorasti.

Roskapostin lähettäminen taas on helppoa siksi, että sähköpostiprotokolla SMTP (Klensin, 2008) ei vaadi minkäänlaista autentikaatiota; kuka tahansa voi väittää olevansa kuka tahansa ja lähettää viestejä mihin tahansa organisaatioon, ainakin jos lähettää viestit ottaen suoraan yhteyttä vastaanottajan postipalvelimelle.

Mitä roskapostituksesta sitten seuraa? Kuten todettua, sähköpostien välittämiseen ja vastaanottamiseen tarkoitetut palvelin- ja verkkoresurssit on mitoitettava

ruuhkahuippujen mukaan, jottei postintoimitus viivästyisi. Koska roskapostitus on usein kampanjaluontoista (Calais et al., 2008; Zheleva et al., 2008), myös ruuhkahuiput ovat usein roskapostituksen seurausta. Varsinaisilla postipalvelimilla roskapostien säilyttäminen kuluttaa myös levytilaa. Vaikka käyttäjät poistaisivat roskapostit saman tien – mitä ei yleensä tapahdu, koska ne usein suodatetaan omiin kansioihinsa – viestit ovat silti moninkertaisen varmuuskopioinnin ja viivästetty poisto -tyyppisten toiminnallisuuksien takia kuluttamassa tallennuskapasiteettia.

Roskapostintunnistusjärjestelmät täytyy rakentaa osaksi sähköposti-infrastruktuuria. Niitä pitää kehittää ja ylläpitää, mikä on kallista. Kaikki mekanismit, kuten palautejärjestelmät ja niihin liittyvät ohjeistukset, pitää organisaatioiden itse rakentaa tai ostaa ulkopuolelta. Yleensä näihin liittyy myös huomattava kuorma mikrotukiorganisaatioille, varsinkin, jos käyttäjäkunta ei ole korostuneen tietotekniikkaorientoitunutta.

Roskapostien poistelu ja omien suodatustoimien hallinnoiminen vie kaikkien sähköpostinkäyttäjien aikaa, erään arvion (Swartz, 2005) mukaan 2.8 minuuttia päivässä. Tuottavuus vähenee, kun roskaposti – siinä missä mikä tahansa sähköposti – keskeyttää meneillään olevan työn. Kaikkia roskaposteja ei ihminenkään suoraan osaa tunnistaa sellaisiksi, mistä seuraa ylimääräistä ajankulua.

Kaikki tämä kuluttaa energiaa ja siten aiheuttaa hiilidioksidipäästöjä, ja koska nykytiedon valossa ihmisen aiheuttamilla hiilidioksipäästöillä on suora vaikutus kasvihuoneilmiön voimistumiseen ja sitä kautta globaalin ilmaston muuttumiseen, energiankulutusta on leikattava. On laskettu, että roskapostin aiheuttamat kasvihuonekaasupäästöt ovat 17 miljoonaa CO_2-ekvivalenttitonnia, joka vastaisi 0.2 prosenttia globaaleista CO_2-päästöistä (McAfee, 2009).

Yksi harvemmin esille nostettu roskapostin haittapuoli on kadotetut viestit. Viestejä tulee virheellisesti leimatuksi roskapostiksi ja siten hylättyä tai unohdettua roskapostikansioon. Myös käyttäjä voi epähuomiossa luulla legitiimiä viestiä roskapostiksi. Tai toisin päin – mikäli roskapostisuodatus on tehotonta, aiheellisia viestejä voi hukkua kohinaan. Roskaposti myös vähentää koko sähköpostin käyttökelpoisuutta ja siten merkittävyyttä esimerkiksi sitä kautta, että sähköpostinkäyttäjien osoitteita on vaikeampi löytää selväkielisenä julkisista lähteistä.

Näiden lueteltujen epäsuorien haittojen lisäksi roskapostista aiheutuu toki myös suoria kustannuksia: roskapostin tuomat haittaohjelmat tai tunnusten kalastelu voi johtaa uusiin tietomurtoihin tai organisaation omilla välineillä toteutettuun roskapostitukseen. Se taas voi vaarantaa koko organisaation sähköpostin toimivuu-

den kolmansien osapuolien ylläpitämien mustien listojen kautta. Toki joskus tulee esiin myös tapauksia, joissa roskapostitse toteutettu huijaus on aiheuttanut suoraa rahanmenoa jollekin uhrille.

2.3 Lainsäädäntö

Roskapostitus on teollistuneissa maissa enimmäkseen kiellettyä, mutta yksityiskohdat vaihtelevat suuresti. Roskapostia säätelevä lainsäädäntö on kuitenkin suuressa osassa maailmaa joko olematonta tai puutteellista, tai – siinä tapauksessa, että lainsäädäntö on olemassa – sen noudattamista ei valvota riittävästi. Näin voi todeta yksinkertaisesti siksi, että roskaposti on edelleen häiritsevä ilmiö.

Kattavan tutkimuksen puutteessa tyydymme toteamaan Wikipedian listan[4] perusteella, että roskapostitusta ylipäänsä säädellään 36 maan lainsäädännössä, joista yhdessä (Bulgaria) roskapostitus on sallittu. Näiden 35 maan joukkoon kuitenkin lukeutuu nekin kymmenen maata, joissa roskapostiongelma on havaittujen tapausten[5] lukumäärän perusteella pahin. Listan kärjessä ovat suurvallat Yhdysvallat, Kiina ja Venäjä. Listasijoituksen järjestys näyttäisi kuitenkin silmämääräisesti korreloivan maiden bruttokansantuotteen[6] kanssa, mikä viittaisi siihen, että roskapostituksen määrä liittyy taloudelliseen toimeliaisuuteen ylipäänsä.

Tarkastelkaamme kahta esimerkkiä, Yhdysvaltoja ja Suomea. Yhdysvaltoja siksi, että se on usein mainittu pahimmaksi roskapostin lähteeksi, ja koska yhdysvaltalaisilla käytännöillä on yleensä vaikutusta yli koko Internetin. Suomea siksi, että suomalainen lainsäädäntö muodostaa vallitsevan toimintaympäristön tämänkin työn kirjoittajalle.

Yhdysvallat sääti 2003 lain, jonka tarkoitus oli hillitä roskapostitusta. Lain nimi, CAN-SPAM (FTC, 2008) ei viittaa siihen, että näin sallitaan roskapostitus – vaikka käytännössä näin näyttäisi käyneen – vaan kyse on pornografian ja harhaanjohtamisen vastaisesta laista, Controlling the Assault of Non-Solicited Pornography and Marketing Act.

Laki määrittää reunaehdot sille, minkälaiset massapostitetut mainosviestit ovat sallittuja. Ehtoja ovat esimerkiksi (FTC, 2008) seuraavat:

[4] `http://en.wikipedia.org/wiki/E-mail_spam_legislation_by_country`, noudettu 17.5.2011.

[5] `http://www.spamhaus.org/statistics/countries.lasso`, noudettu 17.5.2011.

[6] `http://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)`, noudettu 18.5.2011.

- Älä väärennä otsaketietoja (mail headers).
- Älä käytä harhaanjohtavia otsikkoja (subject).
- Ilmoita viestin olevan mainos.
- Kerro oma fyysinen osoitteesi.
- Kerro, kuinka postituslistalta pääsee pois ja noudata omaa ohjettasi.

Laki torjuu siis parhaiten vilpillistä ja harhaanjohtavaa roskapostia, mitä toki suurin osa roskapostista on, mutta ei roskapostia sinänsä. Kriitikoiden mukaan se saattaa tämän vuoksi jopa lisätä roskapostitusta, koska se vain tarjoaa ohjeet ja reunaehdot lailliselle roskapostitukselle (Zhang, 2005). Näin on ilmeisesti myös tapahtunut (Soma et al., 2008). CAN-SPAMin perusteella on toki tuomittu joitakin roskapostittajia ja sen hyväksi puoleksi voidaan myös ainakin yhdysvaltalaisesta näkökulmasta lukea sääntelyn yhdenmukaistamisen yli osavaltionrajojen (Zhang, 2005), mutta se ei näyttäisi vähentävän roskapostia kovin tehokkaasti.

Moni CAN-SPAMin mukainen viesti, joka esimerkiksi sallii poistumisen postituslistalta, on kuitenkin vastaanottajan näkökulmasta roskapostia (Zheleva et al., 2008). Käyttäjähän ei voi tietää, onko lopeta tilaus -linkki viestissä huijausta, jolla varmistetaan vain osoitteen toimivuus, vai ihan todellinen mahdollisuus.

CAN-SPAM on siis niin sanotusti opt out -pohjainen laki eli perustuu jälkikäteisen suostumuksen periaatteseen: roskapostitus on sallittua, kunnes käyttäjä sen eksplisiittisesti kieltää. Suomalainen lainsäädäntö sen sijaan perustuu yksityishenkilöiden osalta etukäteisen suostumuksen periaatteelle (opt in): roskapostitus on sallittua vain, kun lupa on etukäteen kysytty. Laki koskee vain kaupallista roskapostitusta.

Euroopan unionin alueella roskapostia koskevaa sääntelyä on yritetty yhdenmukaistaa direktiivein. Roskapostitusta säädellään erityisesti tietosuojadirektiivillä 95/46/EY ja sähköisen viestinnän tietosuojadirektiivillä 2002/58/EY (Mutchler, 2010). Suomessa on säädetty edellisen pohjalta henkilötietolaki (523/1999) ja jälkimmäiseen liittyen sähköisen viestinnän tietosuojalaki (516/2004).

Henkilötietolaki rajoittaa henkilötietojen käsittelyä, ja sähköpostiosoite katsotaan henkilötiedoksi. Siten esimerkiksi osoitteiden poimiminen www-sivuilta ja siten muodostuneen henkilörekisterin käsittely on laitonta. Sähköisen viestinnän tietosuojalaki taas kieltää lähettämästä mainospostia ilman etukäteistä lupaa. Myös luvan kysyminen erikseen voidaan tulkita tällaiseksi mainokseksi markkinaoikeuden päätöksen (MAO 120/03) perusteella.

Lainsäädännöllinen tilanne on Suomessa siis varsin hyvä yksityishenkilöiden osalta. Valitettavasti tämä koskee käytännössä vain Suomesta lähetettyä kaupallista roskapostia, joka määrällisesti on varsin vähäinen ilmiö. Yritysten ja yhteisöjen kannalta tilanne ei ole näin hyvä: niihin voi kohdistaa suoramarkkinointia, kunnes markkinoinnin kohde sen kieltää. Internetin laajuuden ja pelkästään suomalaisten potentiaalisten massapostittajien määrän huomioiden tämä on varsinkin pienyritysten kannalta kohtuuton ajatus, joka ei oikein skaalaudu: kaikkia ei voi erikseen kieltää roskapostittamasta. Tietosuojadirektiivi olisi sallinut myös opt in -periaatteen soveltamisen oikeushenkilöihin, mutta jätti asian ratkaistavaksi kansallisessa lainsäädännössä, mikä on erikoista ottaen huomioon, että juuri yritykset kärsivät taloudellisia tappioita roskaposti-ilmiöstä (Mutchler, 2010).

Nettiyhteisö ja netissä vallitseva käyttäytymiskoodisto ”netiketti” on usein lainsäädäntöä tiukempi, ja yleensä myös tällainen yrityksille suunnattu roskaposti (engl. B2B spam) todella tulkitaan roskapostiksi. Useimmat sähköposti- tai Internet-palveluntarjoajat (ESP/ISP) ovat pakotettuja kirjaamaan käyttöehtoihinsa ehdottoman roskapostituskiellon ja myös valvomaan sitä. Jos ne eivät tee niin, vaarana on, että koko operaattori joutuu kansainvälisen nettiyhteisön ylläpitämille mustille listoille, joista tarkemmin jäljempänä. Se taas vaarantaisi koko operaattorin ja sen muidenkin asiakkaiden sähköpostinkulun ja kenties muunkin toiminnan. Jotkut palveluntarjoajat ovat immuuneja tälle vaikutukselle, ja tällaisia operaattoreita kutsutaankin roskapostinvastaisissa nettiyhteisöissä nimityksellä *bulletproof hosting*. On kuitenkin huomattava, että sekä CAN-SPAM että tietosuojadirektiivi jättävät tilaa Internet-operaattorien omalle harkinnalle ja antavat mahdollisuuden tiukemmille tulkinnoille. Mikäli näin ei olisi, yksityishenkilöt olisivat vailla suojaa roskapostitukselta (Mutchler, 2010).

Roskapostin suodattaminen ei nimittäin ole myöskään ongelmatonta lainsäädännön kannalta. Ilmeisin ongelma liittyy sananvapauden käsitteeseen, joka Suomen perustuslaissa formuloidaan näin (PL 12 §):

> Jokaisella on sananvapaus. Sananvapauteen sisältyy oikeus ilmaista, julkistaa ja vastaanottaa tietoja, mielipiteitä ja muita viestejä kenenkään ennakolta estämättä. Tarkempia säännöksiä sananvapauden käyttämisestä annetaan lailla.

Onko roskapostien torjuminen sitten sananvapauden rajoittamista? On se. Onko se Suomen perustuslain vastaista? Ei ole. Ensinnäkin, sananvapauden katsotaan

koskevan yksityishenkilöitä, ei yrityksiä tai muita yhteisöjä. Toisekseen, sananvapauden ydinalueella viitataan yleensä nimenomaan oikeuksiin lähettää ja vastaanottaa poliittista viestintää; markkinointi ei nauti yhtä laajaa suojaa. Kolmanneksi, yleensä ajatellaan, että kukin voi tehdä omilla välineillään, mitä haluaa. Tästä seuraa, että yhteisöt voivat itsenäisesti päättää, mitä viestejä ne vastaanottavat verkkoihinsa ja mitä eivät. Tämä ei tietenkään koske kuluttaja-asiakkaille palvelujaan myyviä Internet-operaattoreita, joiden toimintaa säädellään melko tarkasti. Neljänneksi, roskapostin torjunta kuuluu asioihin, joista erikseen säädetään laeissa.

Sähköisen viestinnän tietosuojalain perusteella operaattoreilla ja yhteisötilaajilla, kuten yrityksillä ja oppilaitoksilla, on oikeus torjua roskapostia ja tehdä muita toimenpiteitä tietoturvasta huolehtimiseksi. Tämä oikeus sisältyy pykälään 20:

> Teleyrityksellä, lisäarvopalvelun tarjoajalla ja yhteisötilaajalla sekä niiden lukuun toimivalla on oikeus ryhtyä 2 momentissa tarkoitettuihin välttämättömiin toimiin tietoturvasta huolehtimiseksi:
>
> 1) viestintäverkkojen tai niihin liitettyjen palvelujen tietoturvalle haittaa aiheuttavien häiriöiden havaitsemiseksi, estämiseksi, selvittämiseksi ja esitutkintaan saattamiseksi;
>
> 2) viestin lähettäjän tai viestin vastaanottajan viestintämahdollisuuksien turvaamiseksi; tai
>
> 3) viestintäpalvelujen kautta laajamittaisesti toteutettavien rikoslain 37 luvun 11 §:ssä tarkoitettujen maksuvälinepetosten valmistelun ehkäisemiseksi.
>
> Edellä 1 momentissa tarkoitetut toimet voivat käsittää:
>
> 1) viestin automaattisen sisällöllisen analyysin;
>
> 2) viestien välittämisen ja vastaanottamisen automaattisen estämisen tai rajoittamisen;
>
> 3) tietoturvaa vaarantavien haitallisten tietokoneohjelmien automaattisen poistamisen viesteistä;
>
> 4) muut näihin rinnastettavat teknisluonteiset toimenpiteet.
>
> [...]

Tämä riittää oikeuttamaan lähes kaikki automaattiset tekniset toimet roskapostia vastaan. Internet-operaattorien osalta tätä erikseen jopa vaaditaan (Viestintävirasto, 2008). Mutta mikäli viestien tunnistetietoja halutaan käsitellä ilman tässä

mainittuja perusteita, täytyy soveltaa saman lain pykäliä 13 a–13 k, joille on julkisuudessa annettu nimi lex Nokia. Tämä kuitenkin edellyttää muun muassa ennakkoilmoitusta tietosuojavaltuutetulle ja eräitä muita toimenpiteitä. Tiettävästi mikään yhteisö ei vielä ainakaan kevääseen 2011 mennessä ollut tehnyt vaadittua ennakkoilmoitusta, vaikka kiistelty lakimuutos hyväksyttiin jo vuonna 2009.

2.4 Epätoivoinen kamppailu roskapostia vastaan

Roskapostin torjunta lienee tärkeää paitsi siksi, että siten vähennetään edellä luvussa 2.2 lueteltuja haittoja, myös siksi, että vähentämällä roskapostin perillemenoa ja siten tavoittavuutta, vähennetään myös roskapostituksen kannattavuutta ja siten ennen pitkää myös lähetetyn roskapostin määrää. Tähän lähestymistapaan liittyy kuitenkin myös ongelmia, koska roskapostittaja voi lähes nollakustannuksilla lisätä roskapostituksen määrää torjuttujen viestien tilalle, mikä taas voi kasvattaa roskapostin kokonaismäärää, kuten Klensin (2005) argumentoi. Palaamme tähän ongelmaan myöhemmin luvussa 5.

Historiallisesti on esitetty ja käytetty monenlaisia keinoja roskapostin torjuntaan. Keinot voivat olla teknisiä, lainopillisia tai taloudellisia, ja usein tarvittaisiin ymmärrystä kaikista näistä osa-alueista ainakin enemmän kuin sitä on tarjolla. Välillä on nimittäin esitetty jopa ylilyöntejä ainakin teknisten ja taloudellisten torjuntakeinojen osalta; vuosien varrella on esimerkiksi ehdotettu sähköpostiveroa (esim. Loder et al., 2006). Kiistämättä olisikin hyvä keksiä keino, jolla roskapostiviestin transaktiokustannukset voitaisiin siirtää vastaanottajalta lähettäjälle. Aina välillä (Cranor ja LaMacchia, 1998; Klensin, 2005) muistutetaan kuitenkin perusasioista: roskapostin torjuntakeinoja suunniteltaessa on syytä pidättäytyä sellaisista toimenpiteistä, jotka vaarantaisivat koko sähköpostin olemassaolon käyttökelpoisena mediana.

Käytettyihin ja teknisiin torjuntakeinoihin syvennymme tarkemmin seuraavassa luvussa. Käytössä on kuitenkin mustia listoja, heuristista ja tilastollisiin menetelmiin sekä koneoppimiseen perustuvaa luokittelua, kollaboratiivisia keinoja, maine- ja luottamusjärjestelmiä sekä erilaisia roskapostin ehkäisytapoja. On myös todettu, että roskapostin ja legitiimin postin virrat ovat ihan eri tyyppisiä esimerkiksi koon, vastaanottajien määrän, aikavyöhykkeiden, saapumisprosessin ja muiden piirteiden osalta (Gomes et al., 2007). Miksi siis postilaatikkoon yhä pääsee roskapostia?

Eräs ongelma on teknisten menetelmien epätäydellisyys. Mustista listoista ei saada

kattavia eivätkä oppivat menetelmät voi historiallisen datan perusteella täydellisesti ennustaa tulevaa. Toisaalta roskapostin lähetystä ei saada loppumaan ennen kuin taloudellinen kannustin roskapostitukseen viedään pois ja/tai roskapostitus tulee tarpeeksi voimakkaasti sanktioiduksi.

Oppivat menetelmät vaativat lisäksi paljon laskentatehoa ja tallennuskapasiteettia, mikäli niitä sovelletaan optimaalisella tavalla, eli opettaen niille paikallista ja käyttäjäkohtaista aineistoa. Tällainen lähestymistapa vaatii kuitenkin palvelinkapasiteetin lisäksi suuren määrän työvoimaa, ja jossain voi tulla se raja vastaan, jossa roskapostintorjunnan kustannus tai vaativuus ylittää läpi päästetyistä roskapostista aiheutuvat haitat. Tällaisista syistä teknisesti täydellisyyteen pyrkivää ratkaisua ei käytännössä voida toteuttaa. Sähköpostiympäristössä, jossa käyttäjiä on tuhansia tai kymmeniä tuhansia, helposti kyseeseen tulevat enää niin sanotut yhden koon ratkaisut (one-size-fits-all). Tämä tarkoittaa oppivien menetelmien osalta käyttäjäkohtaisen asemesta globaalia aineistoa eli korpusta, ellei tehokasta hybridiratkaisua löydy. Aineiston problematiikkaan liittyvistä asioista kerrotaan tarkemmin luvussa 4.1.

Tavoitteena olisikin löytää sellainen toimiva roskapostin torjuntakokonaisuus, joka on optimaalinen tai ainakin käyttökelpoinen niiltä osin, kun tarkastellaan esimerkiksi arvioidun tunnistustarkkuuden sekä vaadittujen tietokone- ja henkilöresurssien määrää. Arviot eivät ole koskaan edes otospohjaisesti täsmällisiä, sillä yksityisyyssyistä käyttäjien postilaatikkoja ei voi tarkastella. Roskapostin torjuntakeinojen heterogeenisuus ja osittainen päällekkäisyys on helppo nähdä vahvuutena, mutta yksittäisten keinojen hyödyllisyyttä ja haitallisuutta on hyvä arvioida esimerkiksi juuri edellä kuvatuin perustein.

3 Roskapostin torjuntamenetelmiä

Roskapostia on torjuttu historiassa ja nykypäivänä varsin monenlaisin keinoin, jotka ovat olleet teknisten ratkaisujen lisäksi taloudelliseen kannustamiseen tai rankaisuun perustuvia tai lainopillisia. Tässä luvussa luomme katsauksen käytettyihin ja ehdotettuihin menetelmiin.

3.1 Yleiskuva

Roskapostin teknisiset torjuntakeinot olivat aluksi melko primitiivisiä. Viesteistä saatettiin esimerkiksi etsiä roskaposteille tyypillisiä, erityisen indikatiivisia fraaseja, kuten ”Buy now!” tai ”Click here!”. Tällaisen menetelmän ongelmana oli paitsi sen ylläpitämisen suuritöisyys, myös virhealttius: jos perustaa luokittelun yhteen ainoaan fraasiin, väärien positiivisten todennäköisyys on huomattava. Ennen vuotta 1997 saattoi toimia myös sähköpostiosoitteeseen perustuva mustalistaus: roskapostittajan osoite lisättiin ei-toivottujen lähettäjien joukkoon ja suodatettiin samasta osoitteesta tulevat viestit. Nykyisinhän lähettäjäosoite on aina väärennetty joko kokonaan satunnaisgeneroimalla tai sitten käyttämällä viattomia osoitteita samasta osoitelistasta, joista vastaanottajatkin valitaan.

Vuonna 1997 Paul Vixie ensimmäisen julkisesti levitetyn mustan listan Realtime Blackhole List, jonka perusteella roskapostittajia voitiin estää ottamasta yhteyttä postipalvelimeen (Zdziarski, 2005). Listaa jaettiin aluksi reitittimille BGP-protokollalla, ja sittemmin keksittiin ottaa käyttöön tähänkin soveltuva nimipalvelujärjestelmä, ja näin oli syntynyt ensimmäinen DNS-pohjainen mustalista, DNSBL (Levine, 2010). Niistä tarkemmin luvussa 3.2.

Sääntöpohjainen eli heuristinen suodatus on kehittyneempi versio fraasienmetsästyksestä: siinä etsitään, tyypillisesti säännöllisten lausekkeiden perusteella tiettyjä ominaisuuksia viesteistä, jotka on pisteytetty tietyllä tavalla. Pisteet lasketaan yhteen ja tulos kertoo suodattimen näkemyksen viestin roskapostimaisuudesta. Laajalti käytetty ohjelma SpamAssassin hyödyntää tätä menetelmää, jota selostetaan tarkemmin luvussa 3.7.

Tahallista hidastelua on myös kokeiltu: mitä kauemmin SMTP-transaktio kestää, sitä vähemmän viestejä roskapostittaja ehtii lähettää. Lisäksi tällä keinoin voidaan estää roskapostia tukkimasta palvelimia legitiimiltä postilta. Ohjelmisto nimeltä TarProxy[7] toteuttaa tämän idean, mutta sen kehitys lopetettiin jo vuonna 2004 resurssien puutteessa.

Yhteisöpohjainen suodatus kattaa terminä erilaisia ideoita, joista käytetyimpiä ovat eräänlaisten roskapostien sormenjälkien seuraaminen. Tunnettuja menetelmiä ovat DCC ja Razor/Pyzor. Näistä tarkemmin aliluvussa 3.5.

Roskapostittajalla on yksi heikko kohta hänen yrittäessään saada viestinsä perille: viestin sisältö, hyötykuorma. Mitä hyvänsä temppuja onkin tarjolla, ros-

[7] `http://www.martiansoftware.com/tarproxy/`, noudettu 19.5.2011.

kapostittajan tavoite on kuitenkin saada ”viesti perille”, millä ei nyt tarkoiteta tiettyä roskapostiviestiä vaan sitä tietoa, jonka roskapostittaja haluaa uhrilleen välittää. Roskapostittajalla on rajalliset keinot hyötykuorman naamioimiseen, koska nimenomaan siihen suodattimien on helppo tarttua. Roskapostittaja voi yrittää tahallista väärinkirjoitusta (esim. Hayes, 2007), sisällön piilottamista linkin taakse, sotkemista taustavärin värisellä fontilla, HTML-tageilla tai -kommenteilla, tai aiemmin mainittua kuvaroskapostia. Yhteistä näille hämäämiskeinoille on se, että ne usein toimivat myös itseään vastaan: hyötykuorman perillemeno voi vaarantua ja suodatusohjelmat saattavat tunnistaa viestin helposti roskapostiksi juuri niiden takia.

Tilastolliset ja muut koneoppimiseen perustuvat menetelmät ovat erityisen hyviä tunnistamaan viestistä sekä hyötykuorman että sen piilottamiseen käytettyjä tekniikoita, joita ei sitä paitsi lainkaan esiinny legitiimissä postissa. Niistä tarkemmin seuraavassa luvussa 4.

Joitakin lähettäjän ja viestin validointiin perustuvia menetelmiä, kuten Sender Policy Framework (SPF) ja DomainKeys Identified Mail (DKIM) käytetään rajoitetusti. Näistä tarkemmin aliluvussa 3.4.

Tutkimuskirjallisuudessa on ehdotettu monenlaisia keinoja ja uusia protokollia myös SMTP:n ongelmien korjaamiseksi. Sikäli kun ne eivät rakennu yhteensopivasti nykyisen järjestelmän päälle, niiden käyttöönotto voi olla epärealistista. Vaikka rakentuisivatkin, se edellyttäisi kuitenkin maailmanlaajuiselta nettiyhteisöltä toimenpiteitä, jotka ovat mahdollisesti kalliitakin (Klensin, 2005). Esimerkiksi Duan et al. (2007) ehdottavat artikkelissaan kiinnostavan, SMTP:n kanssa yhteensopivan idean (Differentiated Mail Transfer Protocol) siitä, miten sähköposti voitaisiin muuttaa nykyisestä push-tyyppisestä osin pull-tyyppiseksi, jossa vastaanottaja vastaanottaa vain ne viestit, jotka todella haluaa vastaanottaa. Keino myös edellyttäisi lähettäjän palvelimelta sitoutumista pidemmäksi ajaksi SMTP-transaktioon, mikä ei roskapostittajalle ole mahdollista.

Yllä on kuvattu joitakin käytettyjä ja ehdotettuja keinoja roskapostin *tunnistamiseksi* tai *torjumiseksi*. Esimerkiksi mustien listojen perusteellahan ei varsinaisesti tunnisteta tai luokitella viestejä roskapostiksi vaan torjutaan ne vastaanottovaiheessa lähettäjän IP-osoitteen perusteella ottamatta kantaa sisältöön. Lisäksi on sovellettu erilaisia menetelmiä roskapostin *ehkäisemiseksi*. Nykyisin esimerkiksi on vallitsevana käytäntönä, että henkilökohtaisia sähköpostiosoitteita ei julkaista www-sivuilla selväkielisessä muodossa, josta roskapostittajien hakuautomaatit voisivat

niitä löytää. Tämä ehkäisykeino valitettavasti myös heikentää sähköpostin käytettävyyttä niissä tapauksissa, joissa aiemmin tuntemattomalle ihmiselle pitäisi lähettää postia. Osoitetta voi olla vaikea löytää.

Kun sähköpostin jollain perusteella arvellaan olevan roskapostia, johtopäätöksen jälkeen tehtävät toimenpiteet vaihtelevat. Viesti voidaan joko hylätä SMTP-tasolla, jolloin lähettäjä saa omalta postipalvelimeltaan virheilmoituksen; joissain tapauksissa viesti vain tuhotaan, jolloin lähettäjä luulee viestin menneen perille, mikä voi olla vahingollista väärän positiivisen tapauksessa. Eräs vaihtoehto on hylätä tunnistimen mielestä selvät tapaukset ja ottaa vastaan epäselvät siten, että esimerkiksi niiden otsaketietoihin lisätään merkintä roskapostiepäilystä. Tämän perusteella viestit voidaan postipalvelimella suodattaa omaan roskapostikansioonsa. Näin toimitaan Helsingin yliopistolla, kuten aliluvusta 3.8 käy ilmi.

Seuraavissa aliluvuissa perehdytään tarkemmin eräisiin laajalti käytettyihin ja siten tehokkaiksi havaittuihin menetelmiin.

3.2 Mustat DNS-listat

Mustilla listoilla (DNSBL) useimmiten tarkoitetaan nimipalveluun (DNS) perustuvia estolistoja (blacklists tai blocklists), joilla listataan IP-osoitteita tai osoitealueita. Kyseessä on sähköposti-infrastruktuurin perustekniikka, josta ei sellaisenaan ole tehty paljon tutkimusta, mutta joka itsessään on myös varsin triviaali. Tekniikka rakentuukin olemassaolevien Internet-palvelujen, sähköpostin ja nimipalvelun yhteyteen, eikä teknisesti ottaen sisällä uusia innovaatioita. Sen perustoiminnasta saa hyvän kuvan RFC 5782:n (Levine, 2010) avulla.

Nimipalvelua listojen käytössä hyödynnetään tietojen levittämiseen. DNS-listoja käytetään siten, että postinvälitys- tai roskapostintunnistusohjelmisto kysyy nimipalvelimelta, löytyykö osoite listalta. Palvelin palauttaa tapauksesta riippuen A- tai TXT-tietueen, joka voi sisältää joko IP-osoitteen, kuten 127.0.0.2 tai esimerkiksi syyn listaukseen.

Se, mitkä osoitteet päätyvät listauksiin, riippuu listakohtaisesti vaihtelevista listausperusteista (Jung ja Sit, 2004). Osa listoista on kaupallisia, osa Internet-yhteisön vapaaehtoisesti ylläpitämiä. Helsingin yliopiston käyttämistä listoista merkittävin on Spamhaus ZEN[8], jonka kokemus on osoittanut olevan sekä tehokas että luotettava.

[8] `http://www.spamhaus.org/zen/`, noudettu 28.6.2011.

Yleisesti tiedossa olevia ja usein käytettyjä listausperusteita ovat esimerkiksi seuraavat:

- IP-osoitteessa olevan koneen on havaittu lähettävän postia niin sanottuun roskapostiansaan (*spamtrap*, sähköpostiosoite, johon ei ole tarkoitus lähettää oikeaa postia).
- IP-osoitteen on todettu olevan avoin välityspalvelin.
- IP-osoitteen on todettu kuuluvan dynaamisesti allokoituun tai muuten loppukäyttäjille tarkoitettuun osoitealueeseen.

Mainituista viimeinen listausperuste johtaa erityisen tehokkaaseen roskapostintorjuntaan, koska se ehkäisee tehokkaasti botneteista ja muuten murretuista kotikoneista lähtöisin olevaa roskapostia. Lisäksi nykyisin ollaan laajasti sitä mieltä, että Internet-yhteydellä kytkettyjen kotikäyttäjien ei ole tarpeellista voida lähettää postia suoraan vastaanottajan postipalvelimelle käyttämättä oman operaattorinsa postinvälityspalvelinta. Tästä on olemassa niin kansainvälisiä (MAAWG, 2005) kuin kansallisiakin suosituksia (Viestintävirasto, 2008). Suomen Viestintäviraston määräys sitoo teleoperaattoreita, ja määräyksen toimeenpanolla voidaan arvioida olevan merkittävä vaikutus suomalaisverkoista lähtöisin olevan virus- ja roskapostin vähäiseen määrään.

Jotkut mustalistauksista poistuvat, kun roskapostiongelma on ratkaistu ja koneen ISP ilmoittaa siitä listan ylläpitäjälle. Toiset listaukset ovat pysyviä ja saattavat jopa laajentua koskemaan koko kyseisen osoitteen sisältävää IP-avaruutta, mitä pidetään poliittisena painostuskeinona operaattoria kohtaan, jotta tämä reagoisi verkostaan tulevaan roskapostiin. Toisaalta tämä saattaa johtaa huomattavaan määrään väärin perustein torjuttuja sähköposteja. Listausperusteista käydään Internet-yhteisössä jatkuvaa keskustelua.

DNS-listoja käytetään postinvälitysketjussa ennen sisältöperustaista roskapostintunnistusta, joka on paljon resurssi-intensiivisempää. Jo listojen perusteella voidaan hylätä suurin osa roskapostista, ja tunnistuspalvelimille virtaava postivolyymi muodustuu siten huomattavasti pienemmäksi. Esimerkiksi marraskuussa 2010 noin 60 prosenttia Helsingin yliopistolle pyrkineistä viesteistä torjuttiin SMTP-kättelyssä DNS-listojen perusteella.

Lisäksi DNS-listoja voidaan käyttää myös osana sisältöanalyysiä lukemalla viestin Received-otsakkeita. Esimerkiksi muualta edelleenlähetyspalvelujen – kuten suo-

malainen *iki.fi* – kautta tulleet sähköpostit ”kiertävät” ensimmäisen mustalistatarkistuksen, koska lähettävä IP-osoite, joka on yhteydessä rajapostipalvelimeen, on tyypillisesti luotettu. Tällöin on hyödyllistä tarkistaa myös edelliset välityspisteet, joiden kautta viesti on kulkenut.

Erään erikoistapauksen DNS-listojen joukossa muodostavat niin sanotut URI-listat, jotka sisältävät IP-osoitteiden asemesta roskapostien sisällössä esiintyviä domainnimiä. Roskapostiviestin hyötykuorma, ainoa varsinainen sisältö – siis suodattimien hämäystarkoituksessa lisätyn satunnaistekstin lisäksi – on usein linkki jollekin www-sivulle. Näitä sivuja harvoin voidaan luoda legitiimien www-palveluntarjoajien palvelimille, joten roskapostittajat joutuvat rekisteröimään omia domaineja. Niiden rekisteröinti on taas hieman vaivalloista ja kallista, ja niitä on hyvin rajallinen määrä. Siksi menetelmä on tehokas. Felegyhazi et al. (2010) esittävät artikkelissaan kiinnostavan idean uusien domain-osoitteiden proaktiivisesta listaamisesta. Menetelmä saattaa hyvinkin toimia siksi, että tyypillisesti vain roskapostittajilla on taipumus ottaa uusia domaineita automatisoidusti ja nopeasti käyttöön.

Kolmansien osapuolien tarjoamien DNS-listojen lisäksi lähes kaikilla suurilla sähköpostioperaattoreilla on myös jonkinlaisia yksityisiä estolistoja, joille on listattu esimerkiksi tahoja, joista on tullut runsaasti valituksia, mutta jotka eivät ole päätyneet syystä tai toisesta muille listoille. Lisäksi yksityisillä listoilla voidaan estää yksittäisten häirikköluontoisten IP-osoitteiden postinlähetys.

Tässä luvussa mainitut DNS-listat ovat useimmiten staattisia siinä mielessä, ettei niiden perusteella torjuttu tai torjumatta jätetty viesti vaikuta listojen sisältöön. Olisi mahdollista myös pitää kirjaa roskaposteiksi tunnistettujen viestien lähettäjien IP-osoitteista ja muodostaa näistä yksityinen ja paikallinen musta lista. Sen perusajatus voisi olla, että tietyn roskapostintunnistuksen antaman kynnystason ylittävien roskapostien lähettäjän IP-osoite listataan joillakin perusteilla (esimerkiksi n viestiä m minuutissa). Listauksien kesto voisi myös riippua lähetettyjen roskapostien määrästä. Lähettäville postinvälityspalvelimille annettaisiin pysyvän SMTP-virhekoodin asemesta tilapäinen virhekoodi. Vanhenevien listausten ja tilapäisen virhekoodin ansiosta vältyttäisiin useimmilta vääriltä positiivisilta. Tässä kuvattu idea ei kuitenkaan ole ainutlaatuinen; myös (Cook et al., 2006) ehdottavat vastaavaa menetelmää. Sen ongelmaksi luultavasti muodostuisi esimerkiksi edelleenlähetyspalveluiden, kuten *iki.fi* ja postituslistapalvelinten kautta tuleva roskaposti, jonka seurauksena mainitut legitiimit palvelut voisivat joutua mustalle listalle.

Potentiaalisia vääriä positiivisia voi kuitenkin mustien listojen tapauksessa ehkäistä

käyttämällä niiden rinnalla DNS-pohjaisia valkoisia listoja (DNSWL), joille operaattorit ja muut luotettavaksi katsotut organisaatiot voivat lisätä omia postipalvelinten IP-osoitteitaan. Myös RFC 5782 (Levine, 2010) tuntee tämän mahdollisuuden.

3.3 Harmaalistaus

Harmaalistaus (engl. greylisting) ei nimestään huolimatta ole juurikaan sukua mustille listoille. Sen idea perustuu teoriaan siitä, että roskapostitukseen käytetyt ohjelmat (engl. spamware) eivät sisällä juurikaan virheentarkistusta, eivätkä siten yritä koskaan uudestaan saatuaan SMTP-palvelimelta tilapäisen (4xx) virhekoodin (Levine, 2005). SMTP-protokolla vaatii yrittämään uudelleen, mutta odottamisen ja uudelleenyrittämisen uskotaan olevan mahdollista vain oikeille postipalvelimille; roskapostittajalle arvellaan olevan edullisempaa yksinkertaisesti siirtyä vastaanottajalistalla seuraavaan uhriin.

Harmaalistaus tarvitsee siis tietokannan kaikista niistä tahoista, jotka ovat yrittäneet määritetyn ajanjakson, kuten viikon sisällä lähettää postia. Toteutusten yksityiskohdat vaihtelevat, ja tietokanta-avain voikin olla esimerkiksi IP-osoite tai kolmikko (IP-osoite, lähettäjä, vastaanottaja). Järjestelmän kannattaa suoraan valkolistata suoraan ne avaimet, joiden osalta tiedetään tai muistetaan, että lähettäjä on legitiimi ja siksi yrittää aina uudelleen (Levine, 2005).

Harmaalistausta ehdottanut Harris (2003) listaa tekniikan suunnittelukriteereiksi seuraavia:

1. Mahdollisimman pieni vaikutus käyttäjiin.
2. Rajallinen mahdollisuus estotoimen kiertämiseksi.
3. Mahdollisimman vähän ylläpitovaivaa sekä käyttäjä- että ylläpitäjätasolla.

On kiistanalaista, kuinka hyvin kriteerit täyttyvät harmaalistauksen kanssa. Selvää lienee, että sillä on pystytty toistaiseksi vähentämään jonkin verran roskapostia. Tekniikkaa on kuitenkin myös kritisoitu varsin paljon ja jyrkin sanankääntein (esim. Arment, 2007). Tärkein kritiikin aihe on se, että harmaalistaus aiheuttaa aiheettomia viipeitä postinkulkuun monessa tilanteessa, vaikka toteutus olisikin huolella tehty. Esimerkiksi monet postituslistaviestit, joiden yksilölliseen paluuosoitteeseen (Return-Path-otsake) on koodattu tietoa esimerkiksi lopullisesta vastaanottajasta, saattavat saada yhä uudelleen SMTP-virhekoodia vastaukseksi. Samoin tapauksissa,

joissa postinlähettäjänä toimii palvelinfarmi ja uudelleenyritys voi tulla eri IP-osoitteesta, viesti voi jäädä saapumatta perille kohtuullisessa ajassa.

Näin ollen harmaalistaus todella aiheuttaa melko suuriakin vaikutuksia käyttäjiin. Roskapostittaja voi myös kiertää sen käyttämällä standardinmukaista SMTP-lähetysohjelmaa. Ylläpitovaivaa aiheutuu väärien positiivisten selvittelystä ja manuaalisten valkolistojen ylläpidosta.

Harmaalistauksen haittoja voi tosiaan vähentää yhdistämällä sen käytön paikallisiin ja kolmannen osapuolten ylläpitämiin valkolistoihin (DNSWL), joilta löytyvistä osoitteista tulevat viestit otetaan suoraan vastaan. Myös postituslistaviestit ja muut erikoistapaukset tulisi tunnistaa. Tietokanta-avaimena voisi myös käyttää IP-osoitteen asemesta esimerkiksi sen kolmea ensimmäistä oktettia, jolloin usean IP-osoitteen palvelinfarmit saattaisivat vähän useammin päästä harmaalistauksen ohi.

3.4 Autentikointi ja validointi: SPF ja DKIM

Jos SMTP:n avoimuus on yksi roskaposti-ilmiön perustavia syitä, SMTP:n ohessa toimivat tekniikat Sender Policy Framework (SPF) ja DomainKeys Identified Mail (DKIM) yrittävät korjata ongelmaa. Kyse ei siis ole varsinaisesti roskapostintorjuntatekniikoista vaan tavoista parantaa sähköpostin luotettavuutta, mikä implisiittisesti tarkoittaa roskapostin vähenemistä. Tiedossa tosin ei ole, että roskaposti olisi näiden seurauksena vähentynyt.

SPF (RFC 4408, Wong ja Schlitt, 2006) ja DKIM (RFC 4871, Allman et al., 2007) eivät ole kilpailevia protokollia vaan pikemminkin toisiaan täydentäviä, domaintason autentikointia tarjoavia menetelmiä. SPF:n mukaisesti organisaatio voi julkaista nimipalvelussa (DNS) tietueen, joka kertoo esimerkiksi, miltä palvelimelta saa lähettää viestejä, joiden SMTP-transaktion HELO- ja MAIL FROM -kentissä on mainittu kyseinen domain. DKIM puolestaan allekirjoittaa kryptografisesti lähtevän viestin ja siten varmistaa sen alkuperän kiistämättömästi. Siten voidaan varmistua, että viesti, joka näyttäisi tulevan domainista *example.com*, myös todella tulee sieltä.

SPF:n ja DKIM:n käyttö on viime vuosina lisääntynyt, mutta ei ole kattavaa. Lars Eggertin seurannan mukaan (2011a; 2011b) SPF:ää käyttää globaalisti 61.2% ja DKIM:ää 23% domaineista. Görlingin (2007) mukaan vielä vuonna 2007 vain 1.9% sähköpostia käyttävistä ruotsalaisista .se-loppuisista domaineista julkaisi SPF-tietuetta, ja näistäkin suurimman osan tietue ei sisältänyt käytännössä mitään sellaista, jonka perusteella olisi voinut varsinaisesti tehdä johtopäätöksiä postien

luotettavuudesta.

Tekniikoiden käyttöönottoa hidastaa se, etteivät ne oikein sovellu joka organisaation käyttöön. Niissä on myös joitakin ongelmia. Periaatteellisena ongelmana voidaan pitää sitä, että ne rikkovat Internetin avointa filosofiaa vastaan. Käytännössä siitä seuraa ongelmia esimerkiksi tilanteissa, joissa järjestelmän pitäisi legitiimisti päästä lähettämään sähköpostia toisen puolesta: Esimerkiksi uutispalvelut usein tarjoavat ”Kerro kaverille”-palvelua, jolla voi lähettää kiinnostavan linkin. Kun kuitenkin viestin lähettäjäksi pitäisi ”väärentää” lähettäjän oma osoite, törmätään ongelmiin, jos lähettäjäosoitteen haltijan domain julkaisee SPF-tietuetta. Myös edelleenlähetyspalvelut – kuten Suomessa tunnettu *iki.fi* – sekä postituslistat aiheuttavat ongelmia, mikäli niiden yhteydessä paluuosoitetta (Return-Path) ei kirjoiteta uudelleen. Toisen puolesta ei voi laatia myöskään kryptografista allekirjoitusta.

Yliopiston kaltaisen avoimen organisaation kannalta tekniikoiden ongelma on se, että käyttäjiä ei haluta pakottaa käyttämään välttämättä organisaation omaa postipalvelinta – jonka osoite voitaisiin julkaista SPF-tietueena nimipalvelussa ja joka osaisi DKIM:n mukaisesti allekirjoittaa viestit – koska esimerkiksi operaattorin lähtevän postin palvelimen käyttäminen on joissain tapauksissa helpompaa. Tekniikoiden käyttämisen edellytys olisi se, että kaikki organisaatiosta lähetettävät viestit lähetettäisiin organisaation oman palvelimen – Helsingin yliopiston tapauksessa *smtp.helsinki.fi* – kautta. Tietoturvan kannalta tämä tosin voisi olla myös perusteltua.

Tekniikoiden hyvä puoli on se, etteivät ne vaadi mitään vastaanottajalta tai tämän postinlukuohjelmalta, ja niiden käyttö on erittäin suositeltavaa siellä, missä se on mahdollista, kuten yrityksissä, jotka muutenkin voivat säädellä tarkemmin työntekijöidensä IT-toimintatapoja. Ne ovat hyvä lisä valikoimaan keinoja, joilla lisätään sähköpostin luotettavuutta ja siten vähennetään roskapostia. Erityisesti kalastelua ja muita väärennöksiin perustuvia roskapostituksen lajeja niillä voitaisiin hillitä. Monet suuret palveluntarjoajat, kuten Google, käyttävät SPF:ää ja DKIM:ää menestyksekkäästi osana sähköpostiratkaisuaan (Taylor, 2006; Taylor et al., 2007).

Sähköpostin autentikointiin ja alkuperän luotettavuuden takaamiseen perustuvien menetelmien ilmeinen ongelma on kuitenkin se, etteivät ne kerro mitään viestin sisällön tai lähettäjän luotettavuudesta. Jos esimerkiksi Gmail päästää roskapostittajan luomaan tunnuksia CAPTCHAsta huolimatta, roskapostittaja pääsee lähettämään viestinsä autentikoidusti. Jos validisti autentikoidut viestit vieläpä priorisoidaan postiliikenteessä ja päästetään roskapostitarkistuksen läpi, roskapostittaja voi olla

varma viestinsä perillemenosta.

3.5 Kollaboratiivinen roskapostien tunnistaminen

Kollaboratiivisilla tai yhteisöpohjaisilla roskapostintunnistuskeinoilla viitataan yleensä menetelmiin, joilla lasketaan viesteistä tietyillä algoritmeilla sormenjälki, jota verrataan roskapostiksi ilmoitettujen viestien sormenjälkiin. Sormenjälki voi olla esimerkiksi kryptografinen tiivistefunktio. Mikäli jäljet täsmäävät, voidaan todeta, että kyseessä oli roskaposti. Vipul's Razor (De Guerre, 2007) on eräs tunnettu tällainen järjestelmä; myös DCC[9] (Distributed Checksum Clearinghouses) on melko käytetty.

Toteutusten ja niissä käytettyjen algoritmien yksityiskohdat vaihtelevat, mutta perusajatus on tavalla tai toisella luoda viestistä yksilöllinen tunniste ja siten löytää monelle vastaanottajalle massapostitetut viestit, koska nämä todennäköisesti ovat roskapostia. Idea toimii siksi, että koska roskapostitus on kampanjaluonteista ja jokainen viesti on lähetetty monelle vastaanottajalle, on todennäköistä, että joku muukin on vastaanottanut saman roskapostin (Cormack, 2008). Tähän liittyy kuitenkin myös ongelma: massapostitus ei vielä yksinään tarkoita, että kyse on roskapostista. Esimerkiksi tavalliset, luvalla lähetetyt uutiskirjeet ovat massapostia ja voivat siten näyttää näiden menetelmien valossa roskaposteilta.

Yhteisöpohjaiset roskapostintunnistusmenetelmät ovat hyvä esimerkki siitä yleisestä väitteestä (esim. Crocker, 2005; Klensin, 2005; Görling, 2007), jonka mukaan roskapostin vastainen taistelu on varustelukilpa roskapostittajien ja roskapostituksen vastustajien välillä. Kun ensimmäiset sormenjälkeen perustuvat menetelmät kehitettiin, alkoivat roskapostittajat välittömästi lisätä viesteihin kertaluontoista satunnaisdataa, jonka ansiosta myös sormenjälki muuttuu. Sen seurauksena taas menetelmiä alettiin kehittää sietämään entistä paremmin pientä variaatiota viesteissä (Zhong et al., 2008). Nykyisten sormenjälkiä laskevien algoritmien on osattava tunnistaa luotettavasti melkein-identtiset viestit (Kołcz ja Chowdhury, 2007). Tämä onkin huomattava osa-alue tiedonhakututkimuksessa (Zheleva et al., 2008). On vaikea sanoa, kumpi osapuolista on näiden tekniikoiden osalta kilpailussa edellä; näyttäisi kuitenkin siltä, että nämäkin menetelmät puoltavat paikkaansa yhtenä keinona muiden joukossa: osa roskaposteista kuitenkin tunnistuu niiden avulla. On tosin mahdollista, että kyseiset viestit tunnistuisivat muutenkin roskaposteiksi

[9] `http://www.rhyolite.com/dcc/`, noudettu 12.9.2011

muilla tunnistustavoilla.

Lisäksi kollaboratiiviset menetelmät kärsivät samasta ongelmasta kuin globaaliin (järjestelmänlaajuiseen) aineistoon perustuvat oppivat menetelmät: tavallisten käyttäjien roskapostiraportteihin ei oikeastaan voi luottaa. Tästä lisää luvussa 5.4.

3.6 Haastemenetelmä

Tuon tuostakin ehdotetaan roskapostin torjunnassa käytettäväksi niin sanottua *haastemenetelmää* (engl. challenge/response). Menetelmän idea on, että vastaanottaja kysyy automatisoidusti lähettäjältä jotakin tai pyytää tätä tekemään jotakin ennen kuin vastaanotettava viesti hyväksytään. Menetelmästä on monia sovelluksia erityisesti tunnistautumisen alueella, mutta myös roskapostien torjunnassa sitä on pidetty aivan varteenotettavana vaihtoehtona (O'Brien ja Vogel, 2003; Pelletier et al., 2004; Cook et al., 2006).

Sähköpostin yhteydessä menetelmä toimii esimerkiksi niin, että kun lähettäjä lähettää viestin, vastaanottava postinvälitysohjelma eli MTA (Mail Transfer Agent) panee viestin karanteeniin ja lähettää lähettäjälle jonkin kysymyksen, johon pitää vastata tai tietynmuotoisen WWW-osoitteen, jossa täytyy vierailla.

Vaikka menetelmän rinnalla olisikin valkolistaus käytössä eli tunnetut lähettäjät olisi suoraan sallittu, hidastaa menetelmä silti sähköpostin toimintaa, ja joidenkin mielestä se on paitsi ärsyttävä, jopa vaarallinen (Graham-Cumming, 2005). Haastejärjestelmillä ei suinkaan vähennetä roskapostiviestejä, paitsi ehkä yksittäisen vastaanottajan osalta, vaan päinvastoin luodaan niitä lisää. Haasteautomaatit vastaavat tietysti myös roskaposteihin, joiden lähettäjätiedot on väärennetty, ja niinpä kolmannet, asiaan liittymättömät osapuolet saavat turhia viestejä (Cormack, 2008). Roskapostittajat voivat myös hyväksikäyttää menetelmää triviaalisti samaan tapaan kuin RFC 3834:n (Moore, 2004) mukaisia automaattivastaajia ja saada alun perin hyvämaineiset tahot generoimaan roskaliikennettä. Lisäksi haastejärjestelmät toimivat huonosti postituslistojen kanssa, voivat aiheuttaa postisilmukoita ja ulkoistavat roskapostista aiheutuvan vaivan vastaanottajaorganisaatiolta legitiimille lähettäjälle.

Jotkut DNS-listoja ylläpitävät tahot ovat tiettävästi mustalistanneet myös huonosti konfiguroituja haasteautomaatteja, joiden alkuperäinen tarkoitus on ollut hyvä.

3.7 Sisältöpohjainen analyysi

Sisältöpohjaiseksi analyysiksi tässä nimitetty roskapostintunnistustekniikka ei ole mikään yksittäinen tekniikka, vaan yläkäsite, jonka alle voidaan niputtaa suuri joukko erilaisia roskapostintunnistusmenetelmiä, jotka perustavat päätöksensä nimenomaan sähköpostiviestin sisältöön. Sisältöpohjaisten roskapostintunnistusmenetelmien käyttö alkoi manuaalisista avainsanalistoista. Lisäksi voitiin suodattaa tiettyyn tiedostopäätteeseen – kuten .scr tai .exe – loppuvia liitetiedostoja sisältävät viestit, tietyistä maadomainista peräisin olevat viestit tai esimerkiksi tietyssä formaatissa tai tiettyä merkistöä käyttävät viestit (Cormack, 2008). Tällaisia ad hoc -suodatusmenetelmiä, jotka olivat lähinnä edistyneiden käyttäjien ulottuvilla, saatettiin käyttää vuosituhannen vaihteeseen asti, jonka jälkeen niiden teho ja tarkkuus eivät enää riittäneet.

Kehittyneemmässä versiossa käytetään esimerkiksi säännöllisiin lausekkeisiin perustuvia sääntöjä, joilla etsitään tiettyjä ominaisuuksia viesteistä, joiden tiedetään olevan tyypillisiä roskaposteille tai legitiimeille viesteille, ja näiden sääntöjen mukaan viestille voidaan laskea pisteytys, joka kertoo tunnistimen arvion viestin roskapostimaisuudesta. Laajalti käytetty SpamAssassin-tunnistusohjelmisto toimii näin. Menetelmää voidaan kutsua myös heuristiikaksi (Zdziarski, 2005), ja sen säännöiksi voi toki yhdistää myös muita kuin sisältöpohjaisia menetelmiä, esimerkiksi DNS-listoja ja kollaboratiivisia menetelmiä, joihin tutustuimme luvuissa 3.2 ja 3.5. Sääntöpohjaisen tunnistamisen tärkeä ero yksinkertaiseen avainsanapohjaiseen tunnistamiseen on se, että pisteytyksessä voidaan antaa negatiivisia roskapostipisteitä legitiimeistä ominaisuuksista ja siten vähentää väärien positiivisten todennäköisyyttä.

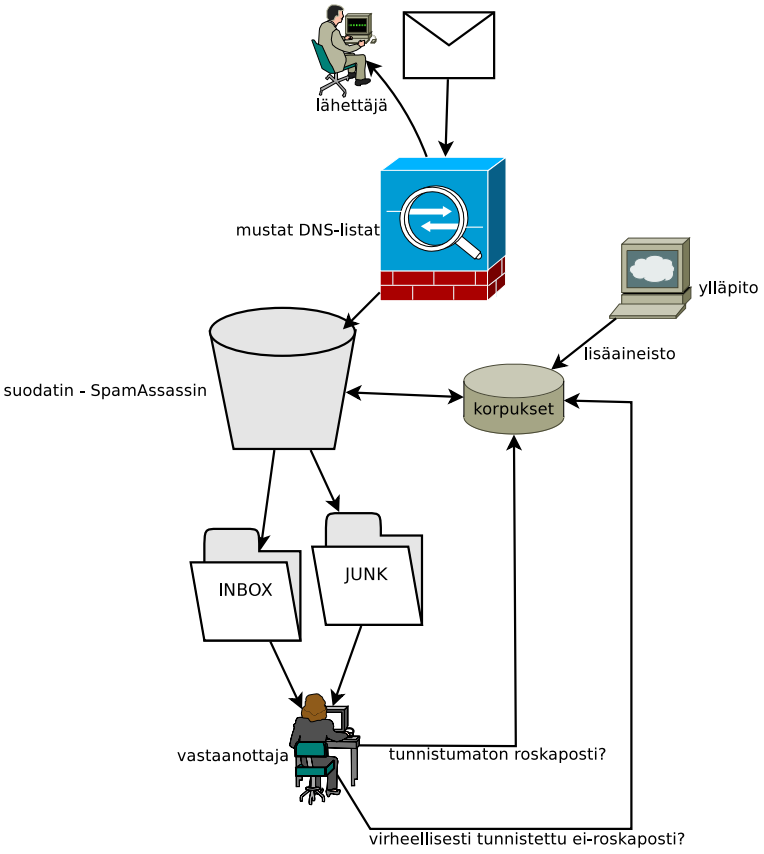
Heuristisen tunnistuksen toinen etu on, että sitä on helppo laajentaa nopeasti tunnistamaan joku tietty roskapostitus- tai kalastelukampanja: tarvitsee vain kirjoittaa uusi tunnistussääntö (Zheleva et al., 2008). Haittapuoli on oikeastaan sama: sääntö pitää kirjoittaa käsin, eikä kukaan ole jatkuvasti reagoimassa uusiin kampanjoihin. Niinpä menetelmä on auttamatta reaktiivinen ja siten usein myöhässä pysäyttääkseen uusia roskaposteja. Käytettävää sääntöjoukkoa on myös aktiivisesti päivitettävä roskapostin evoluution takia (Calais Guerra et al., 2008).

Vuosituhannen vaihteen tienoilla oivallettiin, että sääntöpohjaisia menetelmiä paremmin tulevaisuutta osataan ennustaa tilastollisilla menetelmillä (esim. Androutsopoulos et al., 2000b), joiden tunnistustarkkuus ei enää perustukaan ihmisten kirjoittamiin sääntöihin vaan historialliseen dataan ja sen perusteella laskettuun todennäköisyyteen. Lopputulos on sitä tarkempi, mitä parempaa opetusmateriaalia

tunnistimelle on syötetty, ja mitä parempaa algoritmia se käyttää. Näihin asioihin syvennymme tarkemmin seuraavassa luvussa 4. Sitä ennen kuitenkin tutustumme erääseen roskapostintorjuntajärjestelmän esimerkkitoteutukseen, jossa osa yllä kuvatuista menetelmistä on käytössä.

3.8 Esimerkkitoteutus

Roskapostintorjuntakokonaisuus koostuu useista komponenteista jo sen takia, että monenlaisia menetelmiä tarvitaan toisiaan täydentämään, ja siksi, etteivät yksittäisiä menetelmiä vastaan kohdistetut hyökkäykset läpäisisi koko kokonaisuutta. Kuvassa 2 on loogisella tasolla kuvattu Helsingin yliopiston postijärjestelmä roskapostintorjunnan näkökulmasta.



Kuva 2: Sähköpostijärjestelmän rakenne roskapostintorjunnan osalta.

Organisaation ulkopuolelta lähetetty sähköposti vastaanotetaan palvelimilla, joille on nimipalvelussa luotu MX-tietueet:

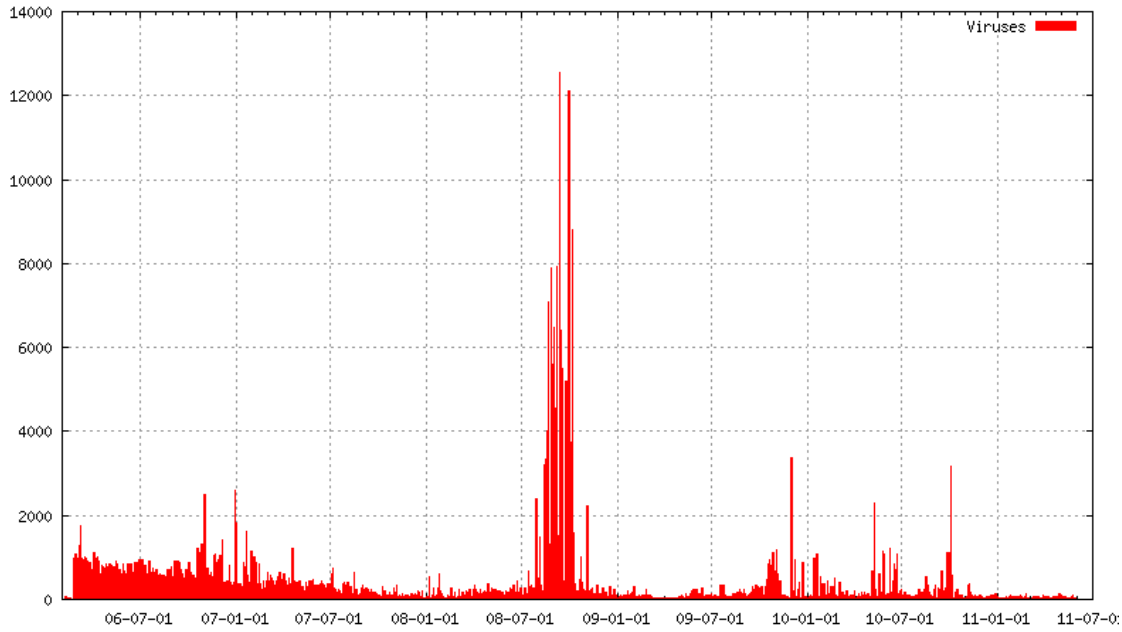
```
$ dig -t mx +short helsinki.fi
20 post.it.helsinki.fi.
20 send.it.helsinki.fi.
20 mail.it.helsinki.fi.
```

Nämä niin sanotut rajapostikoneet pyrkivät hylkäämään ”heti kättelyssä” eli jo SMTP-transaktion aikana mahdollisimman suuren osa vahingollisista viesteistä mahdollisimman vähin kustannuksin eli käyttämällä DNS-pohjaisia mustia listoja, joita vasten verrataan SMTP-transaktion aloittajan IP-osoitetta. Suurin osa roskapostista todella hylätään tässä vaiheessa, ja käyttäjien raportteja vääristä positiivisista ei tämän vaiheen osalta saada, joten valittu lähestymistapa voidaan todeta onnistuneeksi.

Saman SMTP-transaktion aikana tarkistetaan myös vastaanottajan osoite: jo rajapostikoneiden on syytä tuntea omien kohdedomainiensa koko postinimiavaruus eli tietää, mitä osoitteita kussakin postidomainissa on, vaikka palvelinten tehtävä onkin vain välittää postia näille. Mikäli kohdeosoitetta ei ole olemassa, tämä ilmoitetaan jo ennen kuin viesti kuitataan vastaanotetuksi. Jos tämä vaihe ohitettaisiin ja vasta todellinen kohdejärjestelmä osaisi vastata, ettei kyseistä käyttäjää ole, lähettäjälle generoitaisiin virheilmoitusviesti. Roskapostin tapauksessa virheilmoituksen saisi viaton sivullinen, jonka osoite oli väärennettynä lähettäjäksi. Tällaisia viestejä kutsutaan *backscatter*-viesteiksi, ja niitä pidetään yhtenä roskapostien alakategoriana. Legitiimeissä virhetilanteissa – esimerkiksi lähetettyään postia vahingossa väärään osoitteeseen – lähettäjä saa kyllä tällaisen virheilmoituksen, mutta se on tyypillisesti hänen omaan järjestelmäänsä kuuluvan, postia lähettävän postipalvelimen generoima.

Mustien listojen ja niihin rinnastuvien tarkistusten jälkeen saapuva viestimassa ohjataan Helsingin yliopiston tapauksessa Sendmail-postipalvelinohjelmiston milter-rajapinnan kautta virustarkistukseen ja sen jälkeen SpamAssassin-luokitteluohjelmistolle. Nämä molemmat ovat sisältöpohjaisia viestientarkistustapoja ja siten huomattavasti raskaampia, joten on edullista, mikäli mahdollisimman suuri osa viesteistä karsiutuu jo aiemmin pois. Virustarkistus on nykyisin lähes tarpeeton erillisenä komponenttina, jos asiaa arvioidaan lukumäärien valossa: kuvasta 3 ilmenevä kehitys osoittaa, etteivät nykyiset virusmäärät ole enää merkittäviä, vaan pikemminkin häviävät marginaaliin. Taustalla ei suinkaan ole se, että sähköpostitse leviävät virukset olisivat loppuneet, vaan se, että mustat DNS-listat estävät nykyään myös virusten leviämisen tehokkaasti, koska roskapostituksen ammattimaistumisen

seurauksena virukset ja roskaposti leviävät samoista lähteistä ja liittyvät muutenkin kiinteästi toisiinsa.



Kuva 3: Sähköposteista tunnistettujen virusten ja muiden haittaohjelmien määrä per päivä helmikuusta 2006 toukokuuhun 2011.

SpamAssassin luokittelee viestit sääntöpohjaisesti: ohjelman heuristinen luokittelu muodostuu sadoista, viestien ominaisuuksia ja tiettyjä merkkijonoja tunnistavista säännöllisistä lausekkeista sekä liitännäiskomponenteista, jotka tekevät muita, ohjelmakoodina toteutettuja testejä. Näiden joukkoon kuuluu myös bayesiläinen suodatin, joka tunnistaa roskaposteja tilastollisesti sille syötetyn aineiston perusteella. Kaikki testit on pisteytetty koneellisesti: osasta annetaan positiivinen, osasta negatiivinen määrä pisteitä, jotka lasketaan yhteen summaksi, joka kertoo luokittelun tuloksen.

SpamAssassinin luokittelun tuloksena viesteihin lisätään erityiset, luokittelusta kertovat otsakkeet, joiden perusteella käyttäjät voivat lajitella viestit kohdejärjestelmässä tai käyttämässään postiohjelmassa:

```
X-HY-Tests: BAYES_50,RCVD_IN_SORBS_HTTP,RCVD_IN_XBL,RDNS_NONE
X-HY-Spam-Status: LOW ; 53
X-HY-Spam-Level: *****
```

Mikäli SpamAssassinin pisteytyksessä jäädään alle LOW-tason (50), Spam-Status-

ja Spam-Level-otsakkeita ei lisätä lainkaan. Tests-otsake lisätään joka tapauksessa, ja jos käyttäjä haluaa omalla vastuullaan hyödyntää testien nimiä lajittelussaan, se on mahdollista. Tällainen tilanne voi olla esimerkiksi se, jos käyttäjän mielestä liki 100-prosenttisen todennäköisyyden saanut bayesiläisen luokittelun tulos on yksinään riittävä indikaattori roskapostista. Silloin Tests-otsakkeesta löytyy merkintä ”BAYES_99”.

Käyttäjät voivat halutessaan raportoida vääriä positiivisia ja vääriä negatiivisia käyttäen yliopiston webmailia. Näitä raportoituja viestejä käytetään suodattimen manuaaliseen uudelleenopetukseen. Näin saatua aineistoa pitää kuitenkin jonkin verran puhdistaa kaikkein ilmeisimmistä väärin luokitelluista viesteistä. Ja koska käyttäjät tapaavat raportoida käytännössä vain roskapostina pitämiään viestejä, ei-roskaposteja pitää syöttää opetusmateriaaliksi lisäksi satunnaisotannalla.

4 Oppivat tunnistusmenetelmät

Tässä luvussa esitellään sähköpostiviestien sisältöön perustuvia suodatusmenetelmiä, erityisesti oppivaa suodatusta. Roskapostintunnistuksessa on kyse oikeastaan melko tavanomaisesta tekstinluokitteluongelmasta. Erityisen siitä tekee se, että toimintaympäristö on vihamielinen (Xu et al., 2009): aineiston tunnuspiirteet muuttuvat jatkuvasti ja jopa niin, että muutoksen keskeisenä tarkoituksena on luokitteluprosessin hämääminen. Tälle tekstinluokitteluongelmalle on tyypillistä myös muuttujan tilastollisten piirteiden muuttuminen ennustamattomasti, *concept drift* (Cunningham et al., 2003; Delany et al., 2005).

Sahami et al. (1998) ehdottivat naiivin bayesiläisen suodatuksen soveltamista roskapostiongelmaan. Joitakin muitakin artikkeleita julkaistiin jo noihin aikoihin, mutta roskapostintorjunta tilastollisin menetelmin alkoi kuitenkin käytännössä vasta Paul Grahamin (2002) esseestä A Plan for Spam. Hänen kirjoituksensa ei julkaisutapansa tai tyylinsä puolesta edusta perinteistä tieteellistä keskustelua, mutta siihen tehtyjen viittausten ja sen perusteella tehtyjen ohjelmien määrästä voi päätellä, että artikkelin kontribuutio tietojenkäsittelytieteeseen oli merkittävä (esim. Massey et al., 2003; Graham, 2003; Zdziarski, 2005). Guzella ja Caminhas (2009) vertailevat ansiokkaasti Sahamin ja Grahamin ideoita.

Roskapostintorjunnassa puhutaan yleisesti bayesiläisestä suodatuksesta, koska monet Grahamin ehdotuksen mukaan kehitetyt tunnistusmenetelmät löyhästi soveltavat Thomas Bayesin teoreemana tunnettua kaavaa. Perehdymme tarkemmin

aliluvussa 4.3 siihen, kuinka bayesiläinen menetelmä toimii. Myös muita koneoppimiseen perustuvia menetelmiä on tutkittu ja ehdotettu roskapostintunnistukseen; jo Sahami et al. (1998) mainitsevat myös tukivektorikoneet, joita hyödyntämällä D. Sculley sai lupaavia tuloksia väitöstutkimuksessaan (2008). Tukivektorikoneita on toisaalta yleisesti pidetty myös tehottomina, eikä tiedossa ole Sculleyn kevyemmän menetelmän lisäksi kokeita, jotka kumoaisivat käsityksen. Guzella ja Caminhas (2009) analysoivat lisäksi keinotekoiset hermoverkot, logistisen regression ja eräitä muita menetelmiä. Eräissä tutkimuksissa, analyyseissä ja testeissä (Sculley ja Wachman, 2007; Guzella ja Caminhas, 2009; Cormack ja Lynam, 2007; Chang et al., 2008; Qi et al., 2010) juuri logistinen regressioanalyysi on näyttänyt lupaavalta menetelmältä sähköpostien luokitteluun. Sen pitäisi olla nopea (Goodman ja Yih, 2006), tehokas ja hyvin muuttujien tilastollisten piirteiden muutosta (engl. concept drift) ja epätasapainoista aineistoa sietävä (Guzella ja Caminhas, 2009). Itse menetelmään syvennymme tarkemmin aliluvussa 4.4 ja sen soveltamiseen luvussa 6.

Ennen oppimisalgoritmeihin perehtymistä on kuitenkin tutustuttava käytetystä menetelmästä riippumattomiin ja niitä yhdistäviin seikkoihin: seuraavassa aliluvussa (4.1) tutustumme oppivien menetelmien heikkoon kohtaan, oppimateriaaliin ja sen keräämiseen ja aliluvussa 4.2 käymme läpi, kuinka aineisto olisi jäsennettävä ennen prosessointia eli mitä oikeastaan ovat ne sanat tai muut ominaisuudet, joita oppiva menetelmä hyödyntää.

4.1 Oppimisen edellytys: oppimateriaali ja palautteen kerääminen

Tilastollisille ja muille oppiville roskapostintunnistusmenetelmille yhteistä on se, että ne vaativat toimiakseen jonkinlaisen historiallisen aineiston, korpuksen, jota pitää myös päivittää. Kuten todettu, roskapostin tunnistus on vihamielinen luokitteluongelma, koska roskapostin lähettäjät pyrkivät jatkuvasti mukauttamaan viestejänsä niin, ettei tunnistus onnistuisi.

Korpus on jaettava – usein manuaalisesti – kahteen luokkaan, roskapostiin ja legitiimiin postiin, ja oppiva suodatin muodostaa omalla tavallaan käsityksensä siitä, minkälaista kumpaiseenkin luokkaan kuuluva sähköpostiviestintä on. Suodatin tallentaa tietonsa tietokantaan. Koska sähköpostin laatu vaihtelee käyttäjäkohtaisesti, olisi tämän korpuksen syytä olla mahdollisimman henkilökohtainen tai ainakin vastattava tyypillistä käyttöä kyseisessä järjestelmässä (Cormack ja Mojdeh, 2009).

Oppiva luokittelija on siis alustettava autenttisilla, käsin luokitelluilla esimerkeillä siitä, minkälaiset viestit ovat roskapostia ja minkälaiset eivät. Järjestelmää pitää myös päivittää jatkuvasti, jotta se tunnistaisi uudenlaiset roskapostit ja jotta tunnistusvirheistä voisi oppia. Tarvitaan siis palautejärjestelmä, joka sallii käyttäjien syöttää suodattimelle takaisin vääriä positiivisia ja vääriä negatiivisia.

Myös kuvasta 2 hahmottuu tällainen palautejärjestelmä. Ongelma vain on, että käyttäjät eivät toimi, kuten monet tutkimukset tarpeettoman idealistisesti olettavat. Ajatus siitä, että käyttäjät raportoisivat vain aiheettomasti roskapostiksi tunnistuneita viestejä ja virheellisesti tunnistumatta jääneitä roskaposteja vieläpä oikein luokkatunnuksin, on käytännössä osoittautunut epärealistiseksi. Tämä on havaittu myös useissa uudemmissa tutkimuksissa (esim. Sculley ja Cormack, 2008; Cormack ja Kołcz, 2009). Käyttäjäkunta ei edes ole samaa mieltä siitä, mitkä viestit ovat roskapostia ja mitkä eivät. On arvioitu, että jopa 10% käyttäjien raportoimasta palautteesta saattaa olla virheellistä. Esimerkiksi uutiskirjeet raportoidaan helposti roskapostiksi, ja ylipäänsä kaikki mitä käyttäjä ei kenties juuri sillä hetkellä halunnut lukea. On ilmeistä, että tällainen johtaa ongelmiin, mikäli kaikille käyttäjille on yhteinen korpus, jonka perusteella oppiva suodatin tekee luokittelupäätöksiään.

Globaali korpus on kuitenkin tilavaativuudeltaan huomattavasti käyttäjäkohtaista korpusta kustannustehokkaampi (Segal, 2007), koska enemmistö tekstialkioista on kaikille saman organisaation ja saman kielialueen käyttäjille yhteisiä. Ratkaisua voidaan hakea useammasta suunnasta. Esimerkiksi, valinnan ei tarvitse olla absoluuttinen; myös välimuotoja voidaan kehittää. Toisaalta, globaalin tietokannan tapauksessa automaattinen oppiminen tai puolivalvottu oppiminen (engl. semi-supervised learning) ovat vaihtoehtoja. Xu et al. (2009) ehdottavat aktiivista puolivalvottua oppimista juuri SpamAssassinin opettamiseksi.

Voidaan pohtia, olisiko mahdollista järjestää jokaiselle käyttäjälle järjestää oma, henkilökohtainen tietokanta, jolloin vahingossa tai ilkivaltaisesti väärin raportoitujen viestien vaikutus rajoittuisi käyttäjän omaan suodatukseen. Kunkin käyttäjän tallettaman aineiston ja tietokantaindeksien kooksi voitaisiin määritellä esimerkiksi 25 megatavua. Tällöin 50 000 käyttäjän järjestelmässä levyä tarvittaisiin lähes 1.25 teratavua. Määrä ei ole aivan kohtuuton, mutta lisäksi olisi huomioitava, että tietokannasta vain pieni osa voitaisiin pitää keskusmuistissa, ja hitaiden levynoutojen määrä olisi suuri. Järjestelmästä saattaisi tulla siis voimakkaasti I/O-rajoitteinen. Lisäksi globaalia tietokantaa kuitenkin tarvittaisiin ainakin niiden käyttäjien osalta, jotka eivät ole itse raportoineet viestejä. Uudenlaisten roskapostien

opettamisesta koituva hyöty ei myöskään palautuisi koko organisaatiolle.

Käyttäjäkohtaisen tietokannan vaihtoehto on käyttää kokonaan globaalia tietokantaa tilastodatalle. Sen koko voi olla jopa alle 100 megatavua indekseineen, mutta silloin data saattaa niin yleistä, ettei sitä voi pitää ainoana tunnistusmenetelmänä. Joka tapauksessa vaaditun datan määrä on varsin kohtuullinen, jos se on kaikille yhteinen.

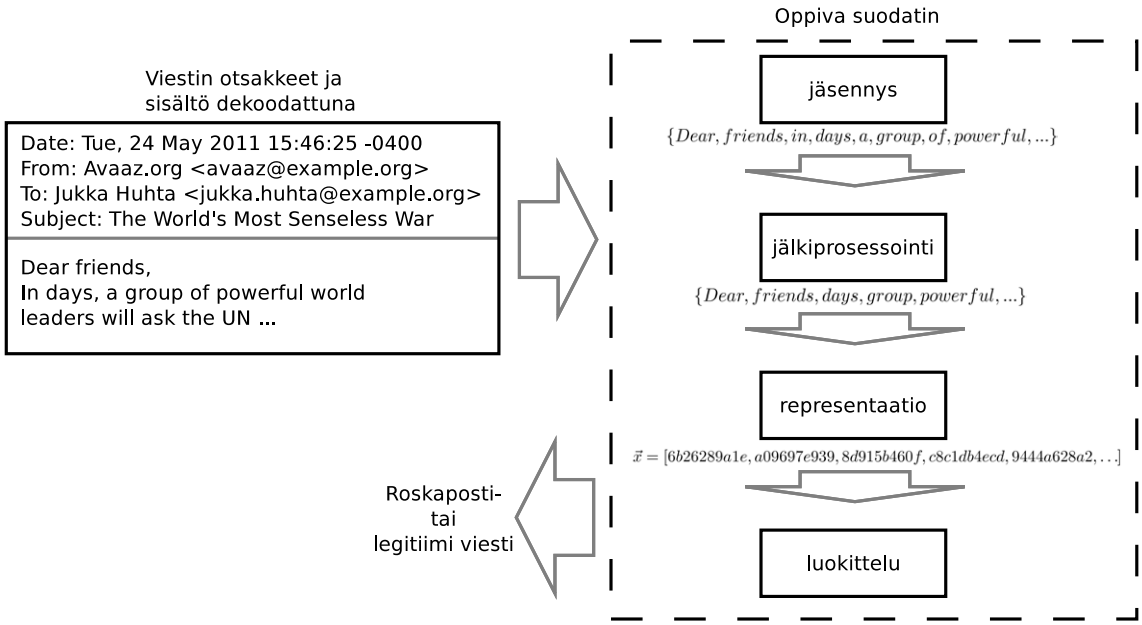
Myös näiden välimuoto on mahdollinen; voidaan esimerkiksi ajatella, että olisi yhteinen 100 megatavun tietokanta, ja sen lisäksi käyttäjäkohtainen tieto siitä, miten kunkin käyttäjän data eroaa tästä keskiarvosta. Erot ovat kuitenkin keskimäärin pieniä, ja viisikin megatavua per käyttäjä saattaisi riittää (Zdziarski, 2005). Sculley ja Cormack (2009) ehdottavat mahdolliseksi ratkaisuksi käyttäjäkohtaista minisuodatinta, joka perustuu tukivektorikoneisiin.

4.2 Mitä oikeastaan opimme: aineiston jäsennys

Puhuttaessa oppivista tekstinluokittelumenetelmistä kiinnostavaa on itse algoritmin lisäksi myös se, mikä on algoritmin käyttämä perusyksikkö, tekstialkio (engl. token). Se voi olla esimerkiksi sana, kuten Grahamin (2003) menetelmässä, monimutkaisempi yhdistelmä merkkejä tai merkkijonoja, kuten n:stä sanasta tai kirjaimesta koostuva n-grammi. Yksi algoritmin tarkkuuteen selvästi vaikuttava tekijä nimittäin on, kuinka aineisto jäsennetään (engl. tokenize) opetus- ja tunnistusvaiheissa. Miten suhtaudutaan esimerkiksi välimerkkeihin? Entä yleisimpiin sanoihin, joita varmuudella esiintyy kummassakin aineistoluokassa? Entäpä kielet, joissa ei ole välilyöntejä samaan tapaan kuin länsimaisissa kielissä? Käsitelläänkö otsakkeita samaan tapaan kuin muita merkkijonoja? Toteutetuissa ratkaisuissa vastaukset edellä mainittuihin kysymyksiin vaihtelevat.

Graham (2002) laski alkuvaiheessa alkioon kuuluvaksi alfanumeeriset merkit, yhdysmerkit, heittomerkit ja dollarimerkit; kaikki muut olivat erottimia. Myöhemmin (2003) hän tarkensi määritelmäänsä muun muassa lisäämällä, että alkuperäinen kirjainkoko tulee säilyttää, ja huutomerkit muuttuvat osaksi alkiota, kuten myös pisteet ja pilkut, mikäli ne ovat kahden numeron välillä esimerkiksi desimaalierottimena. Lisäksi eräät sähköpostin otsakeriveiltä havaitut alkiot merkitään erityisellä tavalla, esimerkiksi ”Subject*Hello”.

Karkeasti ottaen tilastollinen suodatin koostuu käytännössä kolmesta osasta, historiallisen datajoukon ja itse tilastoalgoritmin lisäksi jäsentimestä, joka vastaa



Kuva 4: Oppivan suodattimen keskeiset toiminnot, Guzellaa et al. (2009) mukaillen. Kuvan suodatin noudattelee SpamAssassin-suodatusohjelman bayesiläisen komponentin toimintaa sillä erotuksella, että otsikoita ei käsitellä erikseen kuten ohjelmassa. SpamAssassinin oppivan suodatuksen komponentti jäsentää viestin sisällön lisäksi myös otsakerivit (engl. headers, kuten Subject, User-Agent, Content-Type jne.) ja merkitsee ne erityisin tunnistein.

sähköpostiviestivuon muuttamisesta joukoksi tekstialkioita, joihin sitten algoritmia sovelletaan. Jäsentämisen ja luokittelun lisäksi suodatin tekee eräitä muita asioita, joista kuvassa 4 tarkemmin.

Kuvan sähköpostiviesti on jo valmiiksi dekoodattu ihmisen luettavaan muotoon. Viestin sisältö voi alun perin olla esimerkiksi mielivaltaisella merkistökoodauksella kirjoitettua HTML-koodia, joka on perille välittymisen varmistamiseksi lähettäjän postiohjelmassa vielä koodattu base64-muotoon (ks. RFC 2045, Freed ja Borenstein, 1996). Dekoodattu viesti jäsennetään, pilkotaan osiin sen mukaan, minkälaisia osia käytetty algoritmi pitää alkioinaan. Esimerkiksi Sculley (2008) ehdottaa käytettäväksi neljän merkin n-grammeja, kun taas CRM114-suodatinohjelmisto käyttää alkioinaan usean sanan fraaseja. Sculley tosin käyttää käsitettä k-mer erottaakseen merkkitason n-grammit sanatason vastaavista.

Jäsennyksen jälkeen tai tulkintatavasta riippuen sen osana tekstistä poistetaan niin sanotut *stop wordit* – ne sanat, kuten yleiset artikkelit, pronominit ja vastaavat, jotka esiintyvät taajaan riippumatta aineistoluokasta. Samoin sanat

voidaan palauttaa perusmuotoonsa (Guzella ja Caminhas, 2009). Englanninkielinen tutkimuskirjallisuus (kuten Sculley, 2008; Guzella ja Caminhas, 2009) ei juuri kiinnitä huomiota siihen, että eri kielet toimivat eri tavoin ja monikielisessä ympäristössä toimivien ohjelmien pitäisi kenties osata varautua eri kielten stop wordejä vastaaviin rakenteisiin. Erityisen kieliriippuvaista ja myös summittaista on myös sanojen palauttaminen perusmuotoonsa, kuten Cormack (2008) huomauttaa ja toteaa myös, että tutkimustulokset tämän toimenpiteen hyödyllisyydestä ovat epämääräisiä. Monet suodattimet käsittelevätkin sanoja sellaisena kuin ne esiintyy aineistossa.

Jäsentämisen ja jälkiprosessoinnin jälkeen viestin pitäisi olla syötettävissä oppivalle luokittelijalle joko luokittelua tai oppimista varten. Tosin, esimerkiksi SpamAssassin käyttää tietoalkioinaan raakadatasta lasketun SHA1-tiivistefunktion viittä alinta bittiä, koska siten datan tietokantaesitysmuodon pituus on vakio. Lähteet eivät paljasta, kuinka paljon tämä lisää operaation laskentavaativuutta. Kuvassa 4 on näiden alkioiden heksadesimaaliesitys.

Jäsentimet samoin kuin niitä hyödyntävät tunnistusmenetelmät ja näitä käsittelevä kirjallisuus (esim. Androutsopoulos et al., 2000a) pitävät kyseenalaistamattomana taustaoletuksenaan sitä, että sähköpostissa esitetään lähinnä englanninkielistä, latinalaisilla aakkosilla kuvattua tekstiä. Juurikaan huomiota ei anneta sille tosiseikalle, että sähköposti on kansainvälinen media, ja osa merkittävistä kielistä, kuten mandariinikiina, ei sisällä lainkaan sanan käsitettä. Jäsentimien ja algoritmien soveltuvuus erilaisille kielille voi vaihdella suurestikin.

Zdziarski (2005) huomauttaa, että jäsennin on tilastollisen suodattimen ainoa heuristinen osa, ja sen takia se on pidettävä mahdollisimman yksinkertaisena. Heuristisena ja staattisena komponenttina roskapostinsuodatuksessa se onkin altis erilaisille hyökkäyksille, joita on myös nähty paljon. Hyökkysten perusidea on, että roskapostittaja yrittää estää jäsennintä löytämästä kaikkein paljastavimpia sanoja pilkkomalla niitä joko välilyönnein tai muin merkein tai käyttäen HTML:n sekä JavaScriptin ja CSS:n suomia ulkoasuun perustuvia keinoja (Wittel ja Wu, 2004). Kuvassa 5 on esimerkki tällaisesta välimerkkejä käyttävästä hyökkäyksestä. On tosin melko ilmeistä, että ainakin osa tällaisista yrityksistä muodostuu nopeasti osaksi suodattimen tilastoaineistoa ilmentämään uusien viestien potentiaalista roskapostimaisuutta, mutta joka tapauksessa tällaisten hyökkäyksien jäljiltä alkioiden määrä kasvaa räjähdysmäisesti, kuten Hayes (2007) artikkelissaan osuvasti kuvaa.

```
Date: Thu, 16 Aug 2007 10:23:50 +0200
From: Margie Pfeiffer <Margie.Pfeiffer@southenglandteens.co.uk>
To: jukka.huhta@example.org
Subject: There is simply only one living object.

C Y_T V Take's In-vestor,s F o r Sec.ond Cl'imb!
E_ve+ryone Is Wa+tchin*g C,Y+T V'!

CHIN_A Y+OUTV C+O_R,P ( CYTV*.OB)
$0.4 6 UP F.R*O_M ._3-2 LASTW*EEK


C,Y*T.V con,-tinues i_t_s stea-dy cli.mb f*o+r t h_e seco-nd
w-eek. Sto ck reporti.n_g si_tes acro ss t.h-e boar+d a,r*e issu ing
s_tock wa*tch n,o'tices. R.e.a_d t+h_e new_s, l*o*o*k at
t.h'e number*.s, a.n_d g*e.t on C.Y'T_V as it k-eeps i+t,s c limb g,oing.

BusinessNe+'wsNow h,a,s relea,s,ed C'Y*T V as featu.,red S-tockW-atch.

T.h_i-s o_n,e is s till co+oking_. Go r_e a_d t_h_e n'e-w*s a.n*d g.e.t
on C_Y'T+V Thurs' day 16t'h.!

En'joy t*h_e R-i d,e as we are......

G,O*O*D L,U C,K TRADI+NG AT T*H'E TOP!!'!

[...]
```

Kuva 5: Esimerkki viestistä, jossa yritetään sotkea jäsentimen kyky löytää tekstistä sanoja.

4.3 Naiivi bayesiläinen menetelmä ja muunnelmat

Naiiviksi bayesiläiseksi menetelmäksi kutsutaan tavallisesti luokittelua, jossa sovelletaan suoraan Bayesin sääntöä (Massey et al., 2003):

$$p(S|x) = \frac{p(x|S) \cdot p(S)}{p(x)} \tag{1}$$

Tosin, samaa nimitystä käytetään melko suuresta joukosta menetelmiä, kuten Metsis et al. (2006) artikkelissaan selventävät.

Menetelmää on käytetty tekstiaineistojen luokitteluun jo kauan ennen roskapostiongelmaa, mutta sähköpostin luokittelu on sille hyvä sovellus. Kaavassa $p(S|x)$ on todennäköisyys sille, että viesti on roskapostia sillä ehdolla, että siinä on alkiot x; $p(x|S)$ todennäköisyys, että viestissä on alkiot x ehdolla viesti on roskapostia; $p(S)$ kokonaistodennäköisyys sille, että mikä tahansa viesti on roskapostia, ja $p(x)$ todennäköisyys, että viestissä on alkiot x.

Menetelmää sovellettaessa tehdään yksinkertaistava oletus, että eri alkioiden esiintymistodennäköisyydet ovat toisistaan riippumattomia. Tästä nimitys *naiivi* (esim. Hayes, 2007). Oletus ei päde, koska sähköpostiviestien sisältö ei ole sanastoltaan satunnaista, ja siten voidaan helposti osoittaa, että tietyt sanat esiintyvät tyypillisesti tiettyjen toisten sanojen kanssa; esimerkiksi sanat ”Viagra” ja ”medication” saattavat esiintyä todennäköisesti samoissa viesteissä riippumatta siitä, onko viesti roskapostia vai ei. Tästä väärästä oletuksesta ei kuitenkaan ole haittaa teoriassa (Zhang, 2004), ja on havaittu, ettei myöskään käytännössä (esim. Massey et al., 2003; Robinson, 2003).

Vastaavat todennäköisyydet lasketaan myös käyttäen legitiimiä postiaineistoa H, ja lopulta viesti tulkitaan roskapostiksi, mikäli seuraava epäyhtälö pätee:

$$\frac{(\prod_{i=1}^{m} p(x_i|S)) \cdot p(S)}{p(x)} > \frac{(\prod_{i=1}^{m} p(x_i|H)) \cdot p(H)}{p(x)} \tag{2}$$

Yksinkertaistettuna voidaan sanoa, että naiivi bayesiläinen menetelmä luokittelee viestin roskapostiksi, jos siinä on merkittävästi enemmän ominaisuuksia, joita sen historiallisessa datassa on roskapostista kuin sellaisia ominaisuuksia, joita on esiintynyt legitiimissä postissa (Massey et al., 2003; Androutsopoulos et al., 2000a).

Paul Grahamin (2002) menetelmä on muunnos edellisestä. Olkoot $|S|$ roskapostiviestien määrä historiallisessa aineistossa ja $|H|$ vastaavasti legitiimien viestien määrä. $|S, x_i|$ on sanan x_i esiintymiskertojen lukumäärä aineistossa S ja $|H, x_i|$ vastaavalla tavalla. Mikäli tulo

$$\prod_{i=1}^{m} \frac{\frac{|S,x_i|}{|S|}}{\frac{|S,x_i|}{|S|} + \frac{|H,x_i|}{|H|}} \tag{3}$$

on suurempi kuin 0.9, tulkitaan viesti Grahamin menetelmässä roskapostiksi (Massey et al., 2003; Zdziarski, 2005).

Lisäksi Graham, vääriä positiivia vähentääkseen, päätti tarkoituksellisesti vääristää tuloksia siten, että kertoi kahdella sanan esiintymismäärän legitiimissä aineistossa (Graham, 2002; Zdziarski, 2005).

Robinson (2003) havaitsi, että laskuissa kannattaa ottaa huomioon myös sen historiallisen datan määrä, jolla jonkun sanan roskapostisuusarvo lasketaan, jotta vaikkapa kolme kertaa vain roskapostikorpuksessa esiintynyt sana ei näyttäisi niin ”syylliseltä” kuin huomattavasti useammin esiintynyt. Havainnon tulos on odotettavasti paljon vähemmän vääriä positiivisia luokituksia (Zdziarski, 2005).

Esittäkäämme edellinen kaava toisin. Jokaiselle sanalle w_i roskapostisuustodennä-

köisyys $p(w_i)$ lasketaan seuraavasti:

$$p(w_i) = \frac{\frac{|S,w_i|}{|S|}}{\frac{|S,w_i|}{|S|} + \frac{|H,w_i}{|H|}}. \tag{4}$$

Robinsonin kaavassa

$$f(w_i) = \frac{s \cdot x + n \cdot p(w_i)}{s + n} \tag{5}$$

on kolme muuttujaa, joista n on w_i:n esiintymiskertojen määrä korpuksen *molemmissa* luokissa, x on sanalle annettava lähtöarvo silloin kun $n = 0$, esimerkiksi 0.5, ja s on vakio, jolla painotetaan historiallisen tiedon merkitystä, esimerkiksi 1 (Zdziarski, 2005).

Lisäksi Robinson (2003) ehdotti todennäköisyyksien yhdistämistä käyttäen tilastotieteilijä R. A. Fisherin χ^2-menetelmää. Edellisestä todennäköisyydestä $f(w_i)$ lasketaan yhdistelmätodennäköisyys

$$H = C^{-1}(-2 \log_e \prod_{i=1}^{n} f(w_i), 2n) \tag{6}$$

jossa C^{-1} tarkoittaa käänteistä χ^2-funktiota ja $2n$ vapausasteita. Saatu p-arvo kertoo käytännössä viestille roskapostitodennäköisyyden.

Emme tässä yhteydessä välitä siitä, että saatu todennäköisyys H on todennäköisyys vain, jos olettaisimme nollahypoteesin pätevän. Nollahypoteesi, joka tässä yhteydessä on muotoa ”kaikki $f(w)$ ovat täsmällisiä ja w:t ovat toisistaan riippumattomia”, ei käytännössä koskaan päde, koska, kuten on mainittu, riippumattomuusolettama ei ole pätevä.

Bogofilter-luokitteluohjelman tekijä Louis (2003) vertaili Robinson–Fisher-menetelmää bayesiläiseen ja sai tulokseksi paljon vähemmän virheellisiä luokituksia. Lisäksi Fisheriä käyttämällä ”harmaa alue” selvien tapausten välillä on paljon laajempi, joten menetelmää hyödyntävän suodattimen on mahdollista informoida loppukäyttäjää paremmin epäselvistä tapauksista.

Myös roskapostintorjuntaohjelma SpamAssassin käyttää nykyisin bayesiä ja yhdistelee todennäköisyydet käänteisellä χ^2 -menetelmällä, jota myös O’Brien ja Vogel (2003) pitää tehokkaana. SpamAssassinin bayes-komponentin soveltamiseen perehdytään vertailevasti tarkemmin luvussa 6.

4.4 Logistinen regressioanalyysi

Regressioanalyysiä käyttäen tutkitaan yhden tai useamman selittävän muuttujan vaikutusta selitettävään muuttujaan. Logistinen regressio on tästä erikoistapaus, jossa selitettävä muuttuja voi saada tasan kaksi arvoa. Sähköpostien luokittelussa arvot ovat roskaposti ($Y = 1$) tai ei-roskaposti ($Y = 0$). Logistisella regressioanalyysillä voidaan siis ennustaa todennäköisyyttä sille, että viesti on roskaposti ($P(Y = 1)$).

Myös logistista regressioanalyysiä käyttävässä suodatuksessa sähköpostiviestiaineisto jäsennetään ensin alkioiksi, kuten luvussa 4.2 kuvattiin. Alkiot voivat olla esimerkiksi luonnollisen kielen sanoja tai usean merkin n-grammeja. Joka alkiolle annetaan painoarvo. Olkoot $\vec{w}$ regressiokerrointen eli painotusten vektori ja $\vec{x}$ ykkösten ja nollien vektori, jossa 1 kuvaa sanan löytymistä kyseisestä viestistä. Notaatio $\vec{w} \cdot \vec{x}$ tarkoittaa tällöin sanojen painotusten summaa. Niin kutsuttu logistinen funktio (7) palauttaa luvun väliltä $[0, 1]$, joka on tulkittavissa todennäköisyydeksi $P(Y = 1)$:

$$P(Y = 1|\vec{x}) = \frac{\exp(\vec{w} \cdot \vec{x})}{1 + \exp(\vec{w} \cdot \vec{x})} \tag{7}$$

Jotta viesti voitaisiin luokitella roskapostiksi tämän todennäköisyyden perusteella, määritellään menetelmää sovellettaessa P:lle kynnystaso, esimerkiksi 0.5, jonka ylittävät viestit tulkitaan roskapostiksi. Toinen vaihtoehto on luokitella muuttuja määrätyille luokkaväleille ja pisteyttää luokat, kuten menetelmää SpamAssassinilla suoritettaessa tehdään.

Menetelmästä saadaan ”oppiva” päivittämällä painotuksia *online gradient descent* -menetelmällä. Painokerroin w_i on aluksi nolla, ja sitä päivitetään jokaisen uuden havainnon jälkeen, lisäämällä siihen $(1 - P) \times x_i \times \eta$ jos $Y = 1$ tai vähentämällä siitä $P \times x_i \times \eta$ jos $Y = 0$. Tällöin w_i kertoo, kuinka värittynyt kyseinen alkio on; mikäli kerroin on negatiivinen, alkio on esiintynyt useammin ei-roskapostissa ja päinvastoin.

Lausekkeissa x_i on siis arvoltaan 0 tai 1 riippuen siitä, esiintyykö kyseinen alkio viestissä vai ei. Vakio η taas kuvaa oppimisnopeutta, joka onkin olennainen säätöarvo. Jos luku on liian pieni, oppiminen on hidasta. Jos se taas on liian suuri, oikeaa arvoa jäädään kiertämään siihen kovin hyvin osumatta eli ongelmaksi muodostuu ylisovittuminen, jossa malli kuvaakin paremmin satunnaisvaihtelua kuin muuttujan laatua.

Ylisovittumista voidaan kuitenkin hillitä normalisoimalla painotukset. Vektorin

pituus eli euklidinen normi on määritelty seuraavasti:

$$\|\vec{x}\| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2} \tag{8}$$

Tällöin n-alkioisen vektorin $\vec{x}$ jokainen painokerroin w_i saa säännöllistettynä arvon

$$w_i = \frac{1}{\sqrt{\sum_{i=1}^{n} x_i}} = \frac{1}{\sqrt{n}} \tag{9}$$

koska $\vec{x}$ on binääriarvoinen ja $1^2 = 1$.

D. Sculley (2011) on toteuttanut tässä kuvatun, sinänsä varsin yksinkertaisen menetelmän, ja toteutusta sovelletaan luvussa 6, jotta tietäisimme, ovatko sen tarkkuus ja tehokkuus niin hyviä, kuin edellä, luvun 4 alussa on esitetty.

5 Roskapostintorjunnan ongelmia ja heikkouksia

Edellä on kuvattu roskapostintorjunnassa laajasti käytettyjä menetelmiä. Käytännön kokemukset ovat osoittaneet, että menetelmien kirjo on syytä pitää mahdollisimman laajana. Siten yksittäisiä torjuntatekniikoita vastaan suunnatut hyökkäykset eivät välttämättä läpäise koko suodatusketjua. Tutkimuskirjallisuus tukee tätä käsitystä.

Roskapostintorjunta on kuitenkin jatkuvaa varustelukilpaa roskapostittajien kanssa, eikä ole varmaa, onko kilpa voitettavissa. Globaalissa toimintaympäristössä mikään yksittäinen toimenpide ei kuitenkaan vaikuta kovin radikaalisti, joten organisaation tasolla usein ainoaksi ratkaisuksi jää omien käyttäjien ja tietojärjestelmien suojeleminen roskapostilta mahdollisimman tehokkain keinoin mutta mahdollisimman vähäisin sivuvaikutuksin.

Haasteena on juuri tämän optimipisteen löytäminen. Tässä luvussa pohditaan eräitä roskapostintorjuntaan liittyviä ongelmakohtia, jotka valottavat, miksi luvussa 3.8 kuvattu esimerkkijärjestelmä ei kuitenkaan ole täydellinen.

5.1 Mustat listat

DNS-pohjaisten mustien listojen suurin – joskin alati pienenevä – ongelma on niiden puutteellisuus. Tähän asti on aina löytynyt verkkoalueita, joita ei ole vielä listattu ja joilta roskapostin lähettäminen onnistuu. Vuonna 2011, kun koko IPv4-avaruus

alkaa olla allokoituna, ongelma on pienenemään päin. Epäselvää on, miten IPv6-protokollan käyttöönotto vaikuttaa asiaan.

Mustien listojen kattavuus on toisaalta sukua niiden aggressiivisuudelle. Listojen listauskäytännöt vaihtelevat, ja organisaation tehtävä onkin valita itselleen sopivat listat. Mikäli käytäntönä on esimerkiksi automaattisesti listata roskapostiansoihin (engl. spamtrap) lähetettyjen viestien IP-osoitteita, päädytään helposti listaamaan tunnettuja ilmaispalveluntarjoajia, kuten Hotmail ja Gmail. Tällaiset listaukset aiheuttavat merkittäviä määriä vääriä positiivisia, jos yksittäisen listauksen perusteella hylätään viestejä.

Varsinkin aiemmin mustien listojen ongelmana on ollut juuri kattavuuden puutteesta johtuva reaktiivisuus: IP-osoite voidaan lisätä listalle vasta, kun on todisteet siitä, että osoitetta käytetään roskapostitukseen. Nyt, kun varsinkin Internet-operaattorien tarjoamia laajakaista-avaruuksia on melko kattavasti jo etukäteen listattu[10], ongelma on poistumaan päin. Mutta käsillä onkin toinen ongelma: roskapostittajien on onnistunut automaattisesti rekisteröidä massoittain tunnuksia legitiimeihin ilmaispostipalveluihin, joita ei väärien positiivisten välttämiseksi voida välttämättä listata. Näiden osalta on tukeuduttava sisältöpohjaisiin menetelmiin, kuten heuristiseen analyysiin ja oppivaan tunnistukseen. Näissäkin on ongelmansa, kuten seuraavaksi näemme.

5.2 Heuristiikka

Heuristisen tai sääntöpohjaisen suodatuksen selkärangan muodostavat säännöllisin lausekkein määritetyt säännöt, jotka kuvaavat roskapostien tyypillisiä piirteitä, ominaisuuksia. Jos halutaan torjua roskapostia perustuen aiempien roskapostien ominaisuuksiin – kuten täytyy, jos haluamme nimittää niitä tyypillisiksi ominaisuuksiksi – sääntökokoelma on aina myöhässä ja kuvaa mennyttä aikaa eli on reaktiivinen. Jotta se pysyisi siedettävän tehokkaana, on sitä ylläpidettävä tiiviisti. Harvalla järjestelmäylläpitäjällä on tällaiseen aikaa tai kykyjä. Niinpä tehtävästä käytännössä vastaa ohjelmiston toimittaja tai jokin muu taho, yleensä omaan, pakostikin vajavaiseen aineistoonsa perustuen.

Edellisestä seuraa suoraan, että sääntöjen täytyy olla hyvin yleisiä, koska samoja

[10]Spamhausin Policy Block List (`http://www.spamhaus.org/pbl/`) on tällainen lista; operaattorit voivat itse lisätä sellaisia asiakkaidensa käyttämiä IP-osoitealueita, joista ei pitäisi suoraan lähteä legitiimiä postia.

sääntöjä käytetään monissa organisaatioissa, joilla on heterogeeninen käyttäjäkunta. Lisäksi, jos ja kun säännöt ovat yleisessä jakelussa, myös roskapostittajat voivat käyttää niitä ja testata viestejään kyseisellä suodattimella. Tästä seurauksena roskapostittajat ovat aina askelen edellä heuristisia sääntöjä, joihin yksinomaan perustuvien suodattimien tarkkuus on heikko. Muihin tekniikoihin yhdistettynä saavutetaan toki hyviä tuloksia.

Reaktiivisuuteen ja sääntöjen julkisuuteen on apua oppivista menetelmistä. Nekään eivät valitettavasti ole ongelmattomia.

5.3 Oppivat menetelmät

Näyttää ilmeiseltä, että roskapostintorjuntajärjestelmään on pakko sisällyttää oppiva luokitteluelementti, koska jos jatkuvasti päivittyvää, roskapostin evoluution huomioonottavaa luokittelijaa ei käytetä, suodatuksen tarkkuus heikkenee ajan myötä.

Oppiva sähköpostin luokittelija on kuitenkin roskapostintorjunnan resurssi-intensiivisin komponentti sekä tietokone- että ihmiskapasiteetin määrällä mitattuna. Aineiston jäsennys ja luokittelu kuluttaa suuret määrät laskentatehoa, levytilaa ja I/O-kapasiteettia, eikä SpamAssassinin käyttämä bayesiläinen suodatus tee tästä poikkeusta. Sähköpostien jäsennys (luku 4.2) vaatii paljon kellojaksoja ja muistissa pidettävä, mahdollisesti suurikin tietokanta paljon muistinoutoja. Tilastollista suodatusta tekevien palvelinten muistin määrä onkin oltava suuri. Tietokannan päivitys taas aiheuttaa paljon lukemista ja kirjoittamista. Ongelma on kuitenkin, ettei menetelmien käyttö juuri nopeudu lisäämällä prosessoreja tai vaihtamalla levyjärjestelmä nopeampaan, sillä pullonkaulaksi muodostuu helposti muistinoutojen hitaus (Li ja Zhong, 2006). Niinpä käyttäjäkunnan tai viestimäärän kasvaessa palvelinten määrää onkin lisättävä. Samalla täytyy tietenkin pitää mielessä, että kapasiteetin täytyy riittää myös kuormituspiikkien aikana tai päivitettäessä osaa palvelimista.

Lisäksi oppimateriaalin raportoiminen ja hankkiminen, verifiointi vaatii henkilötyövoimaa. Tai, mikäli se on automatisoitu, väärien positiivisten riski kohoaa. Joka tapauksessa viestien sisältöön perustuvassa oppivassa luokittelussa erehtymisen riski on olemassa, ja riskin suuruus riippuu paljolti opetusmateriaalin laadusta. Siitä tarkemmin seuraavassa aliluvussa. Sisältöön perustuva suodatus on toki virhealtista jo siksi, että määritelmän mukaisesti roskapostissa on kyse suostumuksesta, ei sisällöstä – kuten luvussa 2.1 näimme – ja tietty viesti voikin olla yhdelle roskaposti

ja toiselle ei. Joka tapauksessa paraskin oppiva menetelmä erehtyy aina välillä, ja kuten Mojdeh ja Cormack (2008) kirjoittavat, eri menetelmien tarkkuudesta ei edes tutkimuksessa ole yksimielisyyttä.

Erityisesti bayesiläisen suodatuksen suorituskyky ja tarkkuus jättää siis käytännössä toivomisen varaa. Myös tutkimuskirjallisuus antaa viitteitä tähän suuntaan (esim. Blosser ja Josephsen, 2004; Cormack ja Lynam, 2005). Parempaan tarkkuuteen päästäisiin luultavasti henkilökohtaisilla korpuksilla, mutta ensinnäkin suurelta osalta käyttäjiä ei sellaisten ylläpitäminen onnistu eikä toisaalta sellaista työtä pitäisi käyttäjille sälyttääkään. Toisekseen se taas lisäisi suodattimen vaatimuksia tallennuskapasiteetille, laskentateholle ja I/O-suorituskyvylle. Tutkimuksessa on toki esitelty myös suorituskykyä parantavia ideoita; esimerkiksi Li ja Zhong (2006) ehdottavat tunnistukseen approksimatiivisempaa otetta, mikä vähentäisi laskentatarvetta.

Tukivektorikoneita hyödyntävät suodattimet saattaisivat ratkaista tarkkuusongelman osin, mutta sekä tutkimuskirjallisuus että käytännön toteutusten puuttuminen viittavat siihen suuntaan, että menetelmä on turhan raskas sovellettavaksi käytännössä. Logistisen regression tarjoamiin mahdollisuuksiin tutustumme seuraavassa luvussa.

5.4 Opetusmateriaali

Oppivan suodattimen opetusmateriaalin – siis aidon roskapostin ja legitiimin postin – kerääminen on usein hankalaa ja kallista, kuten Xu et al. (2009) huomauttavat. Tämä on useimmille selvää, mutta siihen tutkimuskirjallisuus on ottanut vain vähän kantaa, että koko oppivien menetelmien teho perustuu sille idealistiselle käsitykselle, että käyttäjiltä saatu palaute vastaisi todellisuutta – siis että raportoidut viestit olisivat luokiteltu oikein. Raportti kokemuksista Gmailin ylläpidossa (Taylor et al., 2007) ja eräät muut tutkimukset (Sculley, 2008; Sculley ja Cormack, 2008; Cormack ja Kołcz, 2009) kuitenkin ottavat asiaan kantaa ja jopa esittävät eräitä korjaustoimenpiteitä.

Ongelma on siis se, että jos käyttäjille tarjotaan toimintoja kuten ”Raportoi roskapostina” ja ”Raportoi ei-roskapostina”, näitä käytetään lähestulkoon satunnaisesti. Roskapostiksi merkitään kaikenlaisia raportointihetkellä ei-toivottuja viestejä riippumatta siitä, täyttävätkö ne mitään yleistä roskapostin määritelmää. Ei-roskapostiksi päätyy viestejä, jotka ovat aivan selviä roskaposteja. Lisäksi ongel-

mana on raporttien yksipuolisuus: ei-roskaposteja ei juurikaan raportoida, paitsi silloin, kun kyse on vääristä positiivisista, jotka ovat melko harvinaisia. Useimmat oppimismenetelmät kuitenkin edellyttävät toimiakseen optimaalisesti suunnilleen samansuuruisen määrän aineistoa kummastakin luokasta. Vinoutunut aineisto on näistä helpompi ongelma; sen voi korjata poimimalla viestejä postilaatikoista satunnaisotannalla. Tietyin hakuehdoin voidaan varmistaa melko suurella todennäköisyydellä, että joukossa ei ole roskaposteja, koska voidaan olettaa, että käyttäjät yleensä poistavat näkemänsä roskapostit.

On havaittu käytännössä, että virheellisesti käsin luokiteltujen viestien opettaminen sellaisenaan oppivalle suodattimelle vääristää oleellisesti suodattimen käsitystä siitä, miten viestejä pitäisi luokitella. Vaikkei kyse luultavasti ole tahallisista väärinkäytöksistä, käyttäjäraportteja on käytännössä pakko – mikäli kyse on yhdestä globaalista aineistosta eikä käyttäjän omasta sähköpostilaatikosta – seuloa ennen opettamista. Seulonta on kuitenkin äärimmäisen vaikeaa, koska koneellisesti viestejä ei voi oikein luokitella – koska niillä oli tarkoitus ojentaa koneellista luokittelijaa – ja täysin manuaalisesti niitä ei voi käsitellä, koska kyseessä on kuitenkin käyttäjien yksityisviestit. On siis pakko hyödyntää jonkinlaista puoliautomaattista seulontaa ja lisäksi anonymisoida viestit sikäli, kun niistä ilmenee henkilötietoja. Tämä kaikki vaatii huomattavia määriä työvoimaa, mutta se on välttämätöntä, mikäli oppivia menetelmiä halutaan käyttää.

Olisi siis syytä kehittää automaattiseen tietojenkäsittelyyn perustuvia ratkaisuja, joilla virheraporttien ongelma voitaisiin kiertää. On myös todettu (Sculley, 2008), että eräät menetelmät, kuten Sculleyn oma tukivektorikone-ehdotus, sietävät melko hyvin yksipuolisia raportteja. Jotkut menetelmät ovat myös robustimpia kuin toiset virheellisesti merkittyjä viestejä vastaan. Väärin merkityt viestit myös jossain määrin häviävät massan joukkoon, mikäli aineiston kokonaismäärä on tarpeeksi suuri. Toisaalta, jos väärin merkittyjen viestien osuus on suunnilleen vakio, niiden määrä toki lisääntyy korpuksen koon kasvaessa.

Yksi ratkaisu voisi olla se, että käyttäjäraportteja hyväksytään vain etukäteen manuaalisesti valituilta ”luotetuilta käyttäjiltä” tai soveltaen esimerkiksi raportoijan maineeseen perustuvaa järjestelyä, jollaisen Zheleva et al. (2008) kuvaavat. Tekijät kuitenkin osuvasti huomauttavat, että kohtalaisen vähäisten käyttäjäraporttien takia luottamusjärjestelmän täytyy olla melko hienostunut, jotta sen avulla saatu aineisto olisi tarpeeksi laaja. Käytännössä raportoijan mainetta voisi siis hyödyntää vasta sitten, kun kyse on riittävän suuresta organisaatiosta tai palveluntarjoajasta,

jonka käyttäjien määrä ylittää jonkin kriittisen rajan. Myös SpamAssassinin bayesiläistä luokittelualgoritmia voisi muokata siten, että niitä data-alkioita, jotka ovat peräisin luotettavaksi havaittujen käyttäjien raportoimista viesteistä, painotettaisiin jollain sopivalla kertoimella.

Lisäksi tulevaisuudessa olisi syytä selvittää, miten aktiivinen puolivalvottu oppiminen (Xu et al., 2009) parantaisi SpamAssassinin suorituskykyä käytännössä. Menetelmässä yhdistellään aktiivista oppimista ja puolivalvottua oppimista klusteroimalla näytteitä, ja alustavat tulokset ovat hyviä.

5.5 Varustelukilpailun päättyminen

Olemme nähneet, että roskapostintorjunta on väistämättä varustelukilpailua, jossa yleensä ollaan vastustajaa askelen jäljessä. Kahdenvälinen kilpa loppuu vain kolmella tavalla (Klensin, 2005):

1. Toinen osapuoli vain luopuu kilvasta,
2. toinen osapuoli pakotetaan ulos kilvasta tai
3. toisen osapuolen rahat tai muut resurssit loppuvat.

Siihen asti, kunnes joku edellä mainituista kohdista täyttyy, jokaiseen toimeen vastataan vastatoimella, joka saa vastaansa vastatoimen vastatoimen. Onko kilpailun loppuminen sitten odotettavissa?

Ensimmäinen kohta on vaikea, koska molemmat osapuolet – roskapostittajat ja roskaposteja torjuvat – ovat kollektiivisubjekteja, joille ei ole mitään yhteistä hallintoa. Koordinoidut toimenpiteet ovat hankalia toteuttaa. Yksittäisen organisaation näkökulmasta jotain on kuitenkin tehdävä, jottei käyttäjien aikaa tuhlaannu liikaa roskaviestien poisteluun. Käytettävä suodatus pitää sitä paitsi olla parempi kuin naapurilla; Klensinin (2005) mukaan roskapostinsuodatuskeinot saattavat vain lisätä roskapostin kokonaisvolyymiä, koska roskapostittaja voi korvata perillemenemättömät viestit lähettämällä uusia. Tämä koituu niiden tappioksi, joiden suodatus ei ole yhtä tiukka.

Toinenkin kohta näyttää hankalalta – emme ole vielä nähneet kattavaa tai sanktioitua ja valvottua lainsäädäntöä, joka vähentäisi oleellisesti roskapostia. Luvussa 2.3 näimme, että sekä USA:n että EU:n yritykset hallita asiaa rikos- tai siviilioikeudellisesti ovat puutteellisia. Vuoden 2011 alusta alkanut roskapostin osuuden

vähentyminen (ks. kuva 1) tosin viittaisi siihen, että toimet ovat kuitenkin saattaneet alkaa purra.

Kolmas kohta on kriittinen yksittäisen organisaation kannalta: missä menee raja, jossa roskapostin torjuntaan käytetyt resurssit ylittävät roskapostin haitoista aiheutuvat kustannukset? Yleensä tätä on organisaation vaikea arvioida, joten roskapostin torjumiseksi tehdään kaikki, mikä on tehtävissä käytettävissä olevin resurssein. Roskaposti ja sen torjunnasta aiheutuvat kustannukset nähdään pakollisena osana viestintää, ja niiden hoitamiseen vain täytyy löytyä voimavaroja, joiden tarve on vain pyrittävä minimoimaan. Mikäli roskapostin määrä todella olisi vähentymässä pysyvästi, myös kannusteet uusien menetelmien käyttöönottoon olisivat vastaanottajaorganisaatioissa vähenemään päin.

6 Logistinen regressioanalyysi sisällönluokittelussa

Edellä kuvattiin eräitä roskapostintorjunnan ongelmia. Kokonaisratkaisua ei liene tarjolla, mutta yksittäisiä ongelmakohtia, kuten bayesiläisen luokittelun epätarkkuutta ja hitautta, voidaan yrittää ratkaista arvioimalla tiettyjen tekniikoiden soveltuvuutta käytettyyn tarkoitukseen ja etenkin kokeilemalla, voisiko parempia tuloksia saada jollain toisella tekniikalla.

Tutkimuskirjallisuudessa (erityisesti Cormack ja Lynam, 2007) ja epävirallisesti (Sculley, 2011) on todettu logistisella regressioanalyysillä saavutettavan hyviä tuloksia roskapostin luokittelussa. Sitä ei kuitenkaan tiettävästi ole käytetty osana torjuntaohjelmistoja, joten tässä luvussa selvitetään, soveltuisiko logistista regressioanalyysia hyödyntävä oppiva komponentti maailman kenties käytetyimmän roskapostintorjuntaohjelmiston, SpamAssassinin, yhteyteen joko olemassaolevan bayesiläistä suodatusta käyttävän suodattimen rinnalle tai sen tilalle. Logistinen regressio on menetelmänä kuvattu tarkemmin luvussa 4.4.

6.1 Koejärjestely ja -aineisto

Logistista regressioanalyysiä – jota jatkossa nimitetään logreg-suodatukseksi – verrataan tässä luvussa muihin vastaaviin suodatusmenetelmiin erilaisin testein, jotka on tarkemmin eritelty taulukossa 1. Tehdyt testit jakaantuvat kahteen ryhmään: toisaalta on verrattu ”puhdasta” logreg- ja bayes-suodatusta kahdella eri korpuksella. Tässä ”puhtaalla” tarkoitetaan sitä, että mukana ei ole muita suodatustulokseen

Taulukko 1: Suoritetut kokeet

Tunniste	Luokittelija	Opetusjoukko
SA_B+L+O.1000	SpamAssassin: bayes- ja logreg-komponentit sekä heuristiset testit	HY-korpus ($n = 2 \times 1000$)
SA_B+L+O.2000	SpamAssassin: bayes- ja logreg-komponentit sekä heuristiset testit	HY-korpus ($n = 2 \times 2000$)
SA_B+L+O.3000	SpamAssassin: bayes- ja logreg-komponentit sekä heuristiset testit	HY-korpus ($n = 2 \times 3000$)
SA_B+O.3000	SpamAssassin: bayes-komponentti ja heuristiset testit	HY-korpus ($n = 2 \times 3000$)
logreg-hycorpus	D. Sculleyn muokkaamaton logistinen regressio	HY-korpus ($n = 2 \times 3000$)
logreg-sacorpus	D. Sculleyn muokkaamaton logistinen regressio	SA-korpus ($n_{Y=1} = 1899$, $n_{Y=0} = 4153$)
bogo-hycorpus	Bogofilter	HY-korpus ($n = 2 \times 3000$)
bogo-sacorpus	Bogofilter	SA-korpus ($n_{Y=1} = 1899$, $n_{Y=0} = 4153$)

vaikuttavia luokittelukomponentteja. Toisaalta on testattu logreg-suodatusta siten, että siitä tehty toteutus on liitetty osaksi SpamAssassinia ja siten päästy vertaamaan sitä SpamAssassinin (versio 3.3.1) omaan bayes-komponenttiin. Tätä vertailua varten on myös suoritettu testejä erikokoisilla korpuksilla, jotta nähtäisiin, miten korpuksen koko (n) vaikuttaa tuloksiin. Tutustumme yksityiskohtiin jäljempänä.

Taulukossa 1 testit SA_B+L+O.1000, SA_B+L+O.2000, SA_B+L+O.3000 ja SA_B+O.1000 ovat SpamAssassin-kokeita. Niissä on käytetty opetusjoukkona tuhannen, kahden tuhannen ja kolmen tuhannen viestin joukkoja siten, että kummassakin luokassa ($Y = 0$ ja $Y = 1$) viestejä on mainittu määrä. Viestit ovat peräisin Helsingin yliopiston postijärjestelmästä (HY-korpus, josta tarkemmin jäljempänä). Tuloksena saadusta raakadatasta on eriteltävissä se, kuinka paljon kukin SpamAssassinin komponentti vaikuttaa tulokseen, joten kumpaakin oppivaa suodatuskomponenttia voidaan suorittaa samalla kertaa, samalle aineistolle. Bayes-

ja logreg-komponenttien lisäksi käytössä on koeasetelman realistisuuden lisäämiseksi myös se osa muista SpamAssassinin heuristisista testeistä, joka ei vaadi verkkoyhteyttä eikä siten perustu reaaliaikaiseen dataan. Kyseessä on siis joukko lähinnä säännöllisin lausekkein määriteltyjä piirteitä, joita etsitään viesteistä. DNS-listat ja kollaboratiiviset menetelmät on poistettu käytöstä, koska ne toimisivat oikein vain reaaliaikaisesti suoritettuna reaaliaikaiselle datalle. Taulukossa lueteltu viimeinen SpamAssassin-testi (SA_B+O.3000) ei sisällä logreg-komponenttia lainkaan. Tämä siksi, että näkisimme, parantaako logreg-komponentin lisääminen suodatustulosta ylipäänsä.

SpamAssassin[11] on laajalti käytetty eri roskapostintorjuntatekniikoita yhdistelevä suodatinohjelmisto, joka on toteutettu vapaana lähdekoodina avoimella Apache-säätiön lisenssillä, ja se on suunniteltu modulaariseksi ja siten helposti laajennettavaksi. Ohjelmointikieli on Perl. SpamAssassinin mukana tuleva, bayesiläisen luokittelun komponentti on toteutettu osin sisäänrakennetusti, osin `Bayes.pm`-liitännäismoduulina. Tässä luvussa käytetty logistisen regressioanalyysin toteuttava moduuli on toteutettu kirjoittamalla `Bayes.pm` vaadituin osin uudestaan ja korvaamalla sen sisältämä bayesiläinen luokittelu D. Sculleyn (2011) alun perin toteuttamalla ohjelmakoodilla[12], josta on tehty SpamAssassin-yhteensopiva versio.

Sekä bayesiläisen luokittelu että logistinen regressioanalyysi palauttavat arvon, joka on tulkittavissa todennäköisyydeksi $P(Y = 1)$ (eli $P(\textit{viesti on roskapostia})$). Arvo on siis tyypiltään jatkuva satunnaismuuttuja väliltä $[0, 1]$. Koska SpamAssassin tekee viestistä kokonaisarvionsa eri tavoin pisteytettyjen testien pistemäärien perusteella, P on luokiteltava diskreeteille väleille. Luokkavälinä on tässä käytetty lukua 0.05.

SpamAssassinin bayes-komponentti tukee ohjaamatonta oppimista siten, että esimerkiksi selviä luokittelutapauksia voidaan syöttää luokittelijalle automaattisesti. Kaikki tällaiset dynaamisesti aineistoa muokkaavat toiminnot on kokeiden ajaksi otettu pois päältä vertailukelpoisuuden varmistamiseksi.

Logreg-suodattimen jäsennysalgoritmi on suurelta osin Sculleyn toteuttama yksinkertainen pilkkoja, koska alkioina käytetään SpamAssassinin sanapohjaisista alkioista poiketen n-grammeja. Tietyiltä osin kuitenkin otsakkeiden jäsennyksessä ja informaatioarvoltaan vähäisten otsakkeiden sivuuttamisessa käytetään SpamAssassinin bayes-komponentin jäsentimen koodia.

Taulukon 1 logreg- ja bogofilter-alkuiset kokeet on tehty siksi, että näkisimme

[11] `http://spamassassin.apache.org/`, noudettu 3.6.2011

[12] `http://www.eecs.tufts.edu/~dsculley/code/perlLogReg.tgz`, noudettu 3.6.2011

”puhtaiden” bayes- ja logreg-suodatusten eron. Logreg-suodattimena toimii täysin muokkaamaton D. Sculleyn (2011) logreg-toteutus, jota verrataan laajalti käytettyyn luokittelijaan Bogofilteriin (versio 0.93.4), joka on puhtaasti bayesiläinen luokittelija. Tällä testillä saadaan vertailukelpoinen tulos siitä, miten logistisen regressioanalyysin tarjoama roskapostin luokittelutarkkuus ja oppimiskyky vertautuu jo valmiiksi käytettyyn tuotteeseen.

Sekä logreg- että bogofilter-kokeiden taustatietokantana on Berkeley DB, jota käytetään Perl-kielen DB_File-moduulin avulla. Siten suorituskykyerojen ei pitäisi johtua ainakaan taustatietokannan suorituskyvystä.

D. Sculley on toteutuksessaan ehdottanut, että viestit katkaistaisiin 2500 merkin kohdalta, muun muassa tietokannan pitämiseksi suhteellisen pienenä. Tämä tietenkin korostaa aina viestin alussa olevien otsaketietojen merkitystä, kenties tarpeettomankin paljon, sillä otsaketietoihin on hyvin helppo tarttua myös heuristisilla menetelmillä ja mustin listoin. Tosin, roskapostit ovat tyypillisesti kooltaan pieniä. Tässä yhteydessä tehdyissä kokeissa katkaisukohta nostettiin kuitenkin 3000 merkkiin.

Logreg-suodattimen muina säätöarvoina on käytetty seuraavia. Merkkitason n-grammien n on 4. Oppimisnopeus $\eta = 0.04$, joka on tuplasti suurempi ehdotettuun (0.02) nähden. Tämä siksi, että todennäköisyysjakauma (ks. kuva 10) olisi selkeämpi. Ominaisuusvektorin normalisointi on käytössä.

Käytetyt aineistot ovat seuraavanlaiset. HY-korpus koostuu käyttäjien raportoimasta roskapostista ja ei-roskapostista. Lisäksi ei-roskapostiluokkaa on täydennetty satunnaisotannalla valituilla aidoilla sähköposteilla. Testeissä käytetty aineisto on otos isommasta ($\approx 2 \times 25000$) joukosta, joita on käytetty tuotantoympäristön roskapostisuodattimen opettamiseen, kuten luvussa 3.8 kuvattiin. Opetusjoukko on pilkottu kolmeen pienempään kaksi kertaa tuhannen (tuhat kummassakin luokassa) viestin joukkoon, jotta näkisimme, mitä oppimis- ja suorituskyvylle sekä tunnistustarkkuudelle käy opetusjoukon kasvaessa. Testausjoukko SpamAssassin-kokeissa koostuu kolmesta tuhannesta viestistä kumpaistakin luokkaa. Kaikki mainitut joukot ovat enimmäkseen erillisiä, mutta valikoitu satunnaisotannalla eri kerroilla, joten joukossa voi olla joitakin samojakin viestejä. Kokeissa käytetyt oppivat suodattimet on opetettu samoilla viesteillä. Logreg- vs. bogofilter-kokeissa on käytetty samoja viestejä testaus- ja opetusjoukkona.

Aineistosta on puoliautomaattisesti ja otsakkeet anonymisoiden karsittu sellaiset näytteet, jotka ovat selvästi virheellisesti luokiteltuja. Joukossa on epäilemättä

väärässä luokassa olevia viestejä, mutta tämä aineiston ominaisuus kenties vain korostaa realistista tutkimusasetelmaa: kyse ei ole ainakaan idealisoidusta suodatusskenaariosta, josta Sculley (2008) väitöskirjassaan kertoo. Aineiston täydellinen siivoaminen ei kuitenkaan ole mahdollista vallitsevan lainsäädännön puitteissa, koska se edellyttäisi käyttäjien viestien manuaalista läpikäyntiä.

SA-korpukseksi tässä kutsutaan julkista SpamAssassin-kehittäjien korpusta[13], jonka luokat ovat huomattavan erikokoiset ($n_{Y=1} = 1899, n_{Y=0} = 4153$), mutta joka on enimmäkseen englanninkielistä. Eri aineistoja verrataan, jotta tietäisimme, missä määrin luokittelutulos kertoo käytetystä opetus- ja testiaineistosta ja missä määrin luokittelijasta.

Kokeet on suoritettu käyttäen Graham V. Cormackin TREC 2006 Spam Evaluation Kitiä[14], joka ajaa annetut roskapostinluokittelutestit annetulla aineistolla ja palauttaa tulokset teksti- ja grafiikkamuodossa. Ohjelmapaketti ei ota kantaa käytettyihin algoritmeihin ja syöteaineistoon; se vain suorittaa roskapostintunnistuksen annetulla aineistolla käyttäen annettua tunnistusohjelmaa ja tuottaa yhteismitalliset tulokset vertailua ja julkaisuja varten.

Laitteistona kokeissa käytettiin Dellin PowerEdge 1955 -korttipalvelinta, jonka prosessori oli Intel Xeon 5140 @2.33 GHz ja jossa oli muistia 4 gigatavua. Palvelimella ei ollut testiajojen aikaan suorituksessa muita prosesseja Linux-käyttöjärjestelmäprosessien lisäksi.

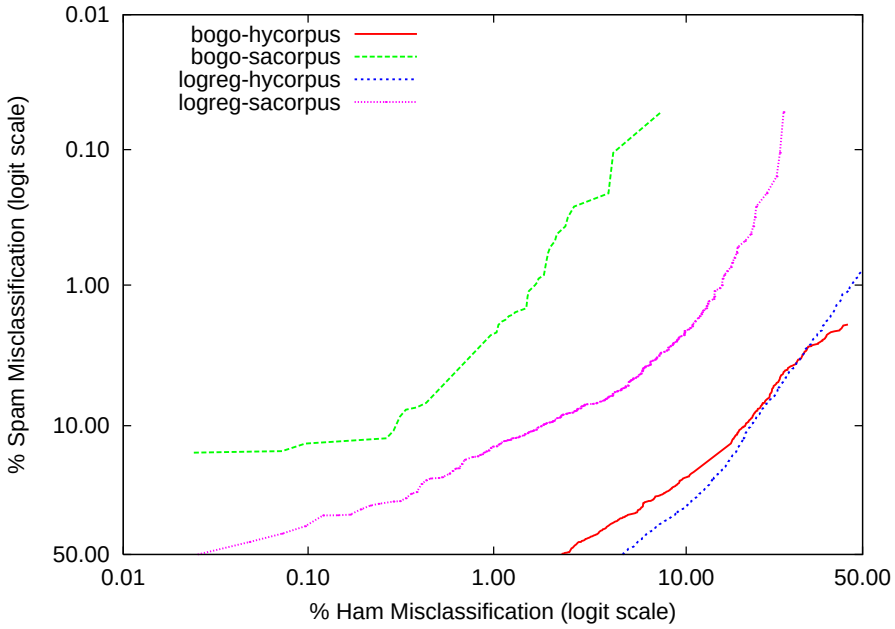
6.2 Tulokset ja päätelmiä

Aluksi on arvioitu muokkaamattoman logreg-suodattimen (Sculley) luokittelukykyä Bogofilter-suodattimeen verrattuna (logreg- ja bogofilter-testit taulukossa 1). Bogofilter on varsin hyvänä pidetty ja laajalti käytetty bayesiläinen luokitteluohjelma. Vertailusta voidaan arvioida ainakin alustavasti, minkä tasoisesta luokittelumenetelmästä logreg-luokittelussa on kyse.

Kuvissa 6 ja 7 Bogofilteriä ja logreg-luokittelijaa on suoritettu sekä HY-korpuksella (2×3000 testijoukko) että julkisella SpamAssassin-korpuksella. Testiajossa ensin luokitellaan viesti ja sitten päivitetään tietokantaa samalla viestillä; testiajo siis simuloi reaaliaikaista, ohjattua oppimista.

Kuva 6 kertoo yllättäen, että tunnistustarkkuus eri luokittelijoilla on ratkaisevasti

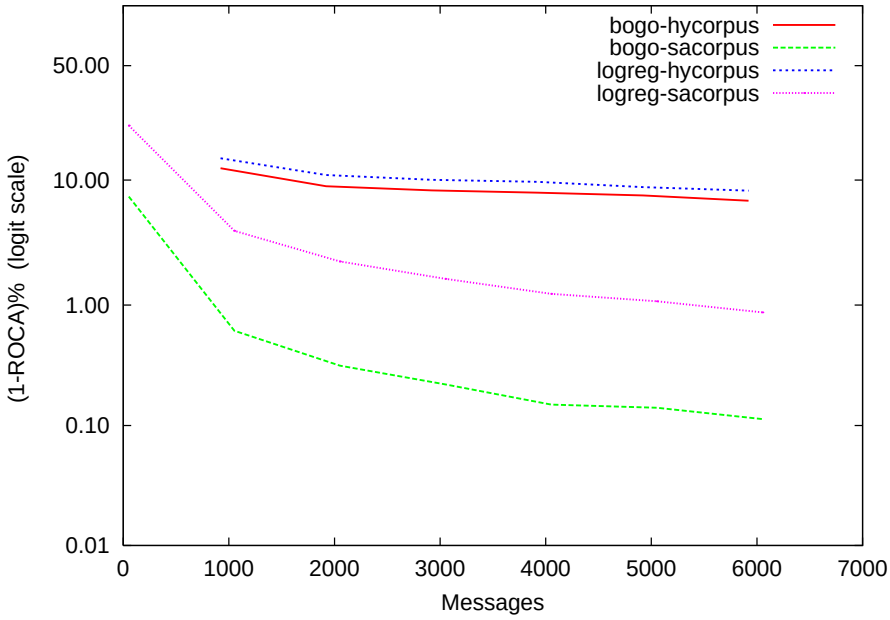
[13] `http://spamassassin.apache.org/publiccorpus/`, noudettu 15.6.2011.
[14] `http://plg.uwaterloo.ca/~gvcormac/jig/`, noudettu 27.6.2011



Kuva 6: Logreg-luokittelijan tunnistustarkkuus verrattuna Bogofilteriin kahdella eri korpuksella.

erilainen riippuen käytetystä korpuksesta: parhaiten selviää Bogofilter SpamAssassin-korpuksella, mutta käytettäessä HY-korpusta ero luokittelijoiden välillä ei ole ollenkaan niin selvä, ja välillä paremmuusjärjestys myös vaihtuu. Kuvasta voi lisäksi päätellä myös sen, että SpamAssassin-korpus on aineistona luokittelijalle paljon helpompi. Sen ei-roskapostiosuus ($Y = 0$) näyttäisikin koostuvan paljolti erään ohjelman kehittäjän henkilökohtaisista viesteistä, kun taas roskapostiluokka on vastaanottajajoukoltaan laajempi. Tällöin jo vastaanottajan käyttämät palvelinten nimet, jotka käyvät ilmi viestiotsakkeista, paljastavat helposti, onko kyseessä roskaposti vai ei. SpamAssassin-korpusta käyttävien luokittelukokeiden hyvistä tuloksista ei siis pitäisi vetää kovin pitkälle meneviä johtopäätöksiä. Aineisto on kuitenkin monissa tutkimuksissa käytetty, joten niiden tutkimusten johtopäätöksiin tulisi kenties suhtautua kriittisesti.

Kuvasta 7 käy ilmi, että molemmilla luokittelijoilla näyttäisi olevan samankaltainen oppimisnopeus eli luokittelutäsmällisyyden parantuminen opetusjoukon kasvaessa. Yllättävän vähän täsmällisyys enää paranee noin kahdentuhannen luokitellun ja opetellun viestin jälkeen, varsinkaan realistisemmalla aineistolla (HY-korpus). Voitaneen kuitenkin todeta, että täsmällisyyden ja oppimiskyvyn osalta logreg-luokittelu on siis ainakin vertailukelpoinen muihin menetelmiin verrattuna.

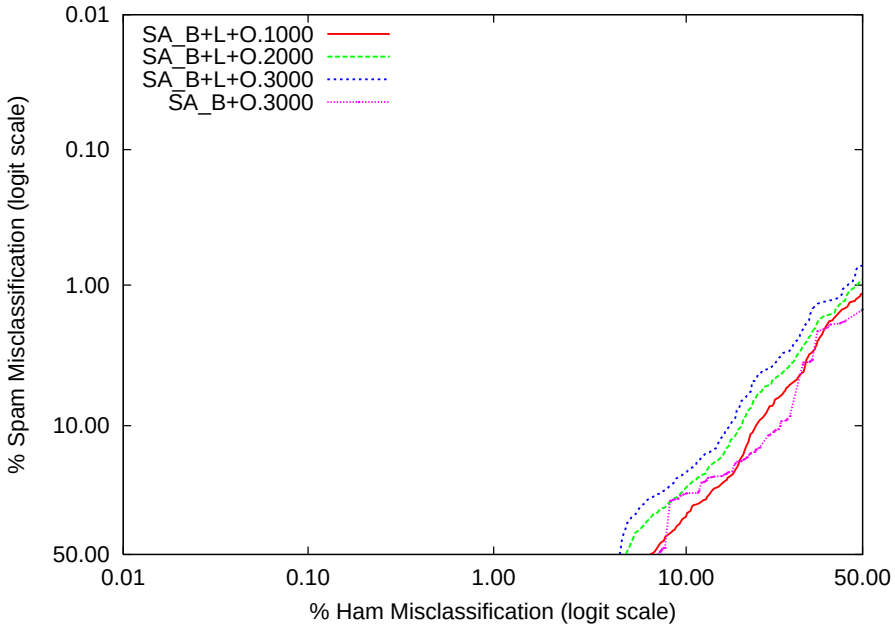


Kuva 7: Logreg-luokittelijan ja Bogofilterin oppimiskäyrän vertailu eri aineistoilla.

Seuraavaksi, kuvassa 8 on vertailtu logreg- ja bayes-komponentteja SpamAssassinista kutsuttuina (SA-alkuiset testit). Mukana on myös muita sellaisia SpamAssassin-testejä, jotka eivät ole kollaboratiivisia tai ulkopuolisten ylläpitämiin palveluihin perustuvia. Sellaisten luotettavuutta olisi nimittäin vaikea jälkikäteen tulkita, kun aineisto ei kuitenkaan kaikilta osin ole aivan tuoretta.

Roskapostien luokittelun täsmällisyyttä vertaillaan usein ns. (1-ROCA)%-mittarilla, joka ottaa hyvin huomioon useampia eri arvoja (Sculley, 2008); luokittelussahan halutaan minimoida väärät positiiviset ja väärät negatiiviset ja maksimoida oikeat. Toimintaominaiskäyrä (Receiver Operating Characteristics Curve, ROC) kuvaa luokittelijan luokittelukykyä koordinaatistossa, josta käy ilmi sekä väärien positiivisten että väärien negatiivisten osuus. ROCA (tai toisissa lähteissä AUC, area under curve) kuvaa käyrän alle jäävää pinta-alaa, ja on siis yhteismitta, joka on ilmaisuvoimainen siitä riippumatta, että jotkut arvottavat vääriä positiivisia eri tavalla kuin toiset. Yksinkertaistetusti: mitä pienempi luku, sitä ”parempi” luokittelutulos.

Kuvasta 8 ja etenkin vastaavista, taulukon 2 ROCA-lukemista on varsin selvästi nähtävissä, että ensinnäkin tunnistustarkkuus paranee merkittävästi opetusjoukon kasvaessa. Testausta voisi toki teoriassa jatkaa loputtomiin kasvattamalla opetusjoukkoa, mutta siinäkin on ongelmansa, kuten jäljempää käy ilmi.



Kuva 8: Bayes, logreg ja muut testit SpamAssassinin osina.

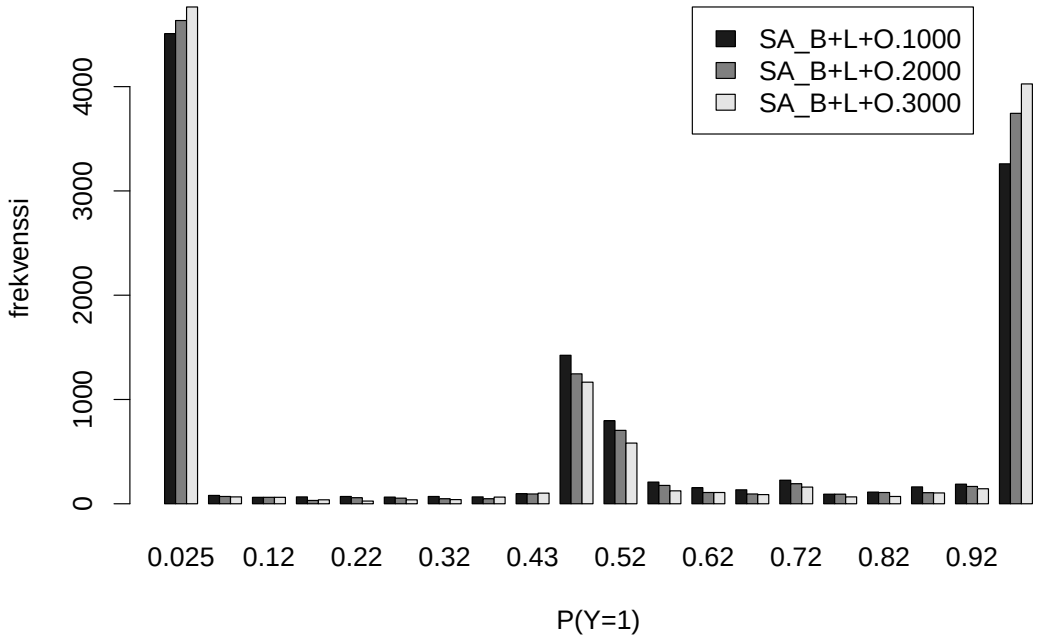
Taulukko 2: SpamAssassin-suoritteisten testien (1-ROCA)%-tulokset

Testin tunniste (sis. opetusjoukon n kummassakin luokassa)	(1-ROCA)%
SA_B+L+O.1000	9.7983
SA_B+L+O.2000	7.7326
SA_B+L+O.3000	6.6814
SA_B+O.3000	9.9546

Toisekseen kuva ja taulukko kertovat vastaanpanemattomasti, että ilman logreg-komponenttia ajettava SpamAssassin osaa luokitella viestejä huomattavasti huonommalla menestyksellä kuin sellainen SpamAssassin, jossa logreg on mukana. Tämä siis puoltaisi logreg-luokittelun käyttöä yhtenä SpamAssassinin testinä.

On kuitenkin todettava, että liki kymmenen prosentin 1-ROCA-lukemat ovat varsin huonoja, eikä sellaisella tunnistustakkuudella voisi kuvitella pärjättävän todellisessa, monen käyttäjän sähköpostiympäristössä, olkoonkin, että paljolti juuri käyttäjäkunnan moninaisuudesta tuloksen heikkous johtuu. Käytännössä kuitenkin opetusjoukot ovat usein paljon isompia, jolloin myös yksittäiset, opetusjoukon väärin luokitellut näytteet häiritsevät vähemmän. Lisäksi pelkkään sisällölliseen tunnistukseen ei tarvitse todellisuudessa nojata, vaan käytettävissä ovat myös esimerkiksi DNS-testit.

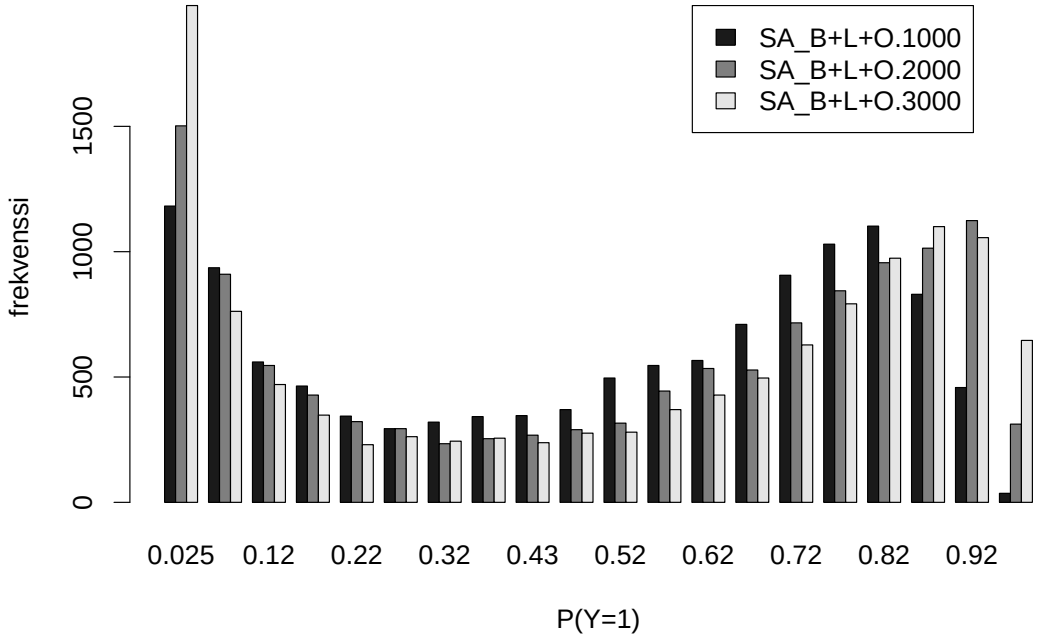
Bayes- ja logreg-todennäköisyyksien jakaumat (kuvat 9 ja 10) sekä niiden erot ovat varsin mielenkiintoisia. Kuvista selviää, että bayes-todennäköisyyksien jakauma painottuu alku- ja loppupäihin sekä hieman aivan lähelle keskikohtaa $P \approx 0.5$. Logreg-jakauma on epäsymmetrisempi mutta myös tasaisempi, ja siinä aivan alkupää ja yläpään osa korostuvat, mutta menetelmä on hieman epävarmempi tulkinnoissaan: lukemia, joissa $P(Y = 1) \approx 1$ ei juurikaan ole, kun opetusjoukko on pieni. Kuitenkin opetusjoukon kasvaessa varmuus lisääntyy tuntuvasti. Jakaumien hahmo käy molemmissa tapauksissa sitä terävämmäksi, mitä suurempi opetusjoukko on kyseessä, mutta erityisesti logreg-todennäköisyyksien varmuus (lähempänä nollaa tai ykköstä) näyttäisi selkeästi kasvavan opetusjoukon kasvaessa.



Kuva 9: Bayes-todennäköisyyksien jakauma erikokoisilla opetusjoukoilla.

Muutoin lupaavanoloisen logreg-luokittelun heikoin kohta näyttää kuitenkin olevan sen nopeus. Tulokset näemme taulukossa 3; opettaminen on jo aluksi varsin hidasta, mutta se vieläpä hidastuu merkittävästi opetusjoukon kasvaessa, jopa niin paljon, ettei opetusjoukkoa oikein voi kasvattaa riittävän suureksi tuotantokäyttöä ajatellen. Kokeessa on siis opetettu ensin 2×1000 viestiä, sitten kahdesti 2×1000 lisää siten, että lopulta opetusjoukon koko on 2×3000.

Pullonkaulana lienevät tietokantaviennit ja -haut, mutta selvittämättä on, miksi



Kuva 10: Logreg-todennäköisyyksien jakauma erikokoisilla opetusjoukoilla.

samankaltaista, Berkeley DB -tietokantaa käyttävät menetelmät käyttävät aikaa noinkin paljon toisistaan poikkeavasti. Lisäksi sekä tietokannan kirjoittaminen että lukeminen hidastuvat merkittävästi opetusaineiston ja siten tietokannan koon kasvaessa. Logreg-kompontentin suorituksen kesto kasvaa lisäksi ainakin aluksi nopeammin kuin lineaarisemmin hidastuvan bayes-komponentin.

Suorituksen osalta tilanne on samankaltainen: jos SpamAssassinilla – joka sisältää 2×3000 opetusjoukolla opetetut bayes- ja logreg-komponentit sekä valmistajan staattiset testit – luokittelee 2×3000 viestin testijoukon, kuluu aikaa peräti 56580 sekuntia. Vastaava testi ilman logreg-komponenttia vie vain 2616 sekuntia.

Vastoin kirjallisuudessa esitettyjä havaintoja logreg näyttäisi siis empiirisesti olevan

Taulukko 3: Oppimisen kesto. Bayes-komponentin opetus on tehtävä luokka kerrallaan.

Opetusjoukko (n)	Käytetty aika (s): bayes	Käytetty aika (s): logreg
2×1000	$70 + 75 = 145$ (100%)	6152 (100%)
2×2000	$96 + 106 = 202$ (139%)	12815 (208%)
2×3000	$121 + 166 = 287$ (198%)	16985 (276%)

liian hidas ja siksi huonosti skaalautuva sovellettavaksi volyymiltään mittavassa tuotantoympäristössä. Erityisen ongelmallinen se on monen käyttäjän ympäristössä, jossa toimitaan globaalin korpuksen pohjalta, koska korpuksen laajuuden täytyisi olla huomattava, jotta se olisi kattava. Osa hitaudesta toki selittyy toteutuskielellä, mutta myös vertailukomponentti on toteutettu Perlillä, joten selitys ei ole riittävä. Säätöarvojen, kuten normalisoinnin poisto tai η-vakion muuttaminen ei vaikuta nopeuteen ratkaisevasti. Viestien katkaisukohdan laskeminen epäilemättä vaikuttaisi, mutta pitäisi erikseen selvittää, mikä olisi tunnistustarkkuuden ja nopeuden suhteen optimaalinen raja.

Saaduista luvuista voisi melko helposti estimoida, kuinka paljon postia päivässä kuvatun kaltaisella laitteistolla voisi käsitellä, mikäli logreg-luokittelua käytettäisiin. Se siis soveltuisi joten kuten pieneen organisaatioon tai henkilökohtaiseen luokitteluun, mutta kovin tehokkaana sitä ei voi pitää.

Lisäksi bayes-luokittelu vaikuttaisi sietävän logreg-luokittelua paremmin vinoutunutta aineistoa. Logreg vaatii käyttämänsä menetelmän (gradient descent) takia, että näytteet ovat jotakuinkin satunnaisessa järjestyksessä; sille ei siis käy bayesista poiketen se, että opetettaisiin luokallinen näytteitä kerrallaan, koska logreg-algoritmi ei sisällä mitään mekanismia, joka tasapainottaisi vinoutunutta näyteaineistoa.

Mikäli hitausongelman saisi ratkaistua, logreg-komponentin voisi hyvin ottaa tuotantokäyttöön yhtenä SpamAssassinin testinä, joka täydentäisi bayesiläistä luokittelua varsin hyvin. Tässä käytettyä referenssitoteutusta, jonka ominaisuudet riittivät näihin testeihin, mutta ei varsinaiseen tuotantoon, täytyisi täydentää ja hioa monin tavoin. Ainakin seuraavia parannusehdotuksia pitäisi harkita nopeuttamisen lisäksi:

- Opetetuista viesteistä tulisi pitää kirjaa, jolloin samoja viestejä ei tulisi ainakaan epähuomiossa opetettua uudelleen.
- ”Epäoppiminen” tai unohtaminen pitäisi toteuttaa; on tosin mahdotonta samaan tapaan kuin bayes-luokittelussa, koska tietokannan päivittäminen vaikuttaa aina senhetkisiin painotuksiin. Toki viestin voi opettaa uudelleen toisella luokkatunnuksella.
- Jäsentämistä voisi myös optimoida lukemalla raakadataa sellaisenaan dekoodaamatta base64- tai quoted-printable-koodausta ja tulkitsematta HTML:ää – kuten Qi et al. (2010) ehdottavat – jos käytettäisiin tavutason n-grammeja. Tällöin voitaisiin louhia piirteitä sellaisenaan myös binäärisistä liitetiedostoista, jotka voivat olla esimerkiksi PDF-, kuva- tai jopa MP3-tyyppisiä.

- Logreg- ja bayes-menetelmien yhdistäminen hybridimalliksi, kuten Chang et al. (2008) ehdottavat.

7 Yhteenveto

Roskapostiongelma näyttäisi edelleen vaivaavan nettiyhteisöä, organisaatioita ja yksittäisiä käyttäjiä, eikä lopullista ratkaisua siihen ole näköpiirissä. Aikojen saatossa on monenlaisia keinoja ehdotettu ja kokeiltu. Taloudellisissa keinoissa yritetään jakaa kustannustaakkaa oikeudenmukaisemmaksi siirtämällä roskapostin lähettämisen transaktiokustannuksia vastaanottajilta lähettäjille. Tiedossa ei ole, että tällaiset keinot olisivat jossain tulleet käyttöön.

Lainopillisin keinoin roskapostikamppailu tarkoittaa roskapostituksen tai jonkin siihen liittyvän toimenpiteen, kuten osoitelistojen keräämisen kriminalisointia tai rajoittamista. On olemassa lainsäädäntöä, esimerkiksi Suomessa, joka on osittain onnistunut tehtävässään, mutta silloinkin valvonta ja sanktiointi ovat puutteellisia.

Teknisiä keinoja on suuri valikoima, ja tässä työssä on esitelty niistä merkittävimpiä. Esimerkiksi mustat DNS-listat auttavat tehokkaasti ja luotettavasti roskapostin torjunnassa jo varhaisessa vaiheessa, jolloin ei suurelta osalta viestimassaa jouduta lainkaan resurssi-intensiiviseen sisältöpohjaiseen viestien tarkistukseen. Sisältöpohjaisessa tarkistuksessakin on useita menetelmiä, mutta tutkimuksessa eniten huomiota ovat kiinnittäneet koneoppimiseen pohjautuvat menetelmät, osin siksi, että koneoppiminen on muutenkin oma alueensa, ja koska siltä kentältä on edelleen odotettavissa viestien koneelliseen luokitteluun parannuksia.

Erityisesti tutustuimme perinteisen bayesiläisen sähköpostinluokittelun ja logistiseen regressioanalyysiin perustuvan luokittelun eroihin teoriassa ja käytännössä. Osoitimme, että tarkkuuden osalta menetelmissä ei ole suurtakaan eroa, mutta luokittelijat kiinnittävät huomiota hieman erilaisiin asioihin ja niiden palauttama todennäköisyysjakauma on erilainen. Täten menetelmät täydentävät hyvin toisiaan, ja niitä voisi hyvin soveltaa rinnakkain esimerkiksi SpamAssassin-luokitteluohjelmasta kutsuen. Logistinen regressio ei ainakaan testatulla toteutuksella ollut kuitenkaan tarpeeksi nopea, ja toteutusta pitäisikin saada nopeutetuksi merkittävästi, jotta se olisi käytännössä sovellettavissa.

Empiiristen kokeiden perusteella vaikuttaa siltä, että koneoppimista soveltavissa menetelmissä oleellisinta onkin käytetty aineisto, jonka laadun varmistamiseen kannat-

taa kiinnittää huomiota paljon enemmän kuin käytettyjen algoritmien, menetelmien ja niiden parametrien hiomiseen. Esimerkiksi hyvällä käyttöliittymäsuunnittelulla, käyttäjien koulutuksella ja mainepohjaisuutta hyödyntävillä palautejärjestelmillä voitaisiin edistää roskapostien ja ei-roskapostien raportointia niin, että niitä raportoitaisiin mahdollisimman vähän väärillä luokkatunnuksilla.

Koneoppimismenetelmien lisäksi on siis syytä tutkia ja käyttää mahdollisimman monimuotoista keinovalikoimaa, jotta roskapostittajien temput eivät auta viestejä pääsemään kaikista seulan osista läpi. Uusien torjuntamenetelmien kanssa on kuitenkin aina muistettava, että sellaiset ratkaisut eivät ole hyväksyttäviä, jotka unohtavat sähköpostin toiminnallisuudesta olennaisia asioita lyhytaikaisten roskapostinvastaisten hyötyjen toivossa. Sellaiset menetelmät taas, jotka vaativat globaalin infrastruktuurin muutosta, eivät vain tule otetuksi käyttöön (Crocker, 2005; Klensin, 2005).

Toisaalta, pelkkä tekninen roskapostintorjunta ei riitä, koska mitä tehokkaampia keinoja käytetään, sitä enemmän roskapostittajat lisäävät volyymiä, mikäli yhteiskunta ei lähetä tarpeeksi voimakasta signaalia siitä, että toiminta ei ole hyväksyttävää. Volyymin kasvattaminen sattuu eniten niiden nilkkaan, joiden suojaus ei ole yhtä hyvä kuin muilla. Sääntelyn kiristäminen lienee siis välttämätöntä, jotta kilpavarustelu saadaan pysäytettyä.

Välillä näkyy kuitenkin valonpilkahduksia. Kuvasta 1 näkyvä vuoden 2011 pudotus roskapostin suhteellisessa osuudessa voisi kertoa siitä, että varustelukierre olisi hetkeksi loppunut ja jotkut roskapostittajat olisivat purkaneet linnoituksiaan tai antautuneet. Toivokaamme, että tila on pysyvä eikä väliaikainen.

Lähteet

Allman, E., J. Callas, M. Delany, M. Libbey, J. Fenton ja M. Thomas, DomainKeys Identified Mail (DKIM) Signatures. RFC 4871 (Proposed Standard), updated by RFC 5672, 2007.

Androutsopoulos, I., J. Koutsias, K. V. Chandrinos, G. Paliouras ja C. D. Spyropoulos, An evaluation of naive bayesian anti-spam filtering. *Proc. Workshop on Machine Learning in the New Information Age*, sivut 9–17, 2000a.

Androutsopoulos, I., J. Koutsias, K. V. Chandrinos ja C. D. Spyropoulos, An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. *Proc. 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, sivut 160–167, ACM Press, New York, NY, USA, 2000b.

Arment, M., Greylisting: The worst thing to happen to email since spam. Saatavilla: `http://articles.marco.org/238`, noudettu 24.5.2011, 2007.

Blosser, J. ja D. Josephsen, Scalable centralized bayesian spam mitigation with bogofilter. *Proc. 18th USENIX conference on System administration*, sivut 1–20, USENIX Association, Berkeley, CA, USA, 2004.

Bradbury, D., Botnets behind spam surge. *Network Security*, *2006*(12), 2006.

Calais, P. H., D. E. V. Pires, D. O. Guedes, W. Meira, C. Hoepers ja K. Steding-jessen, A campaign-based characterization of spamming strategies. *Proc. 5th Conference on Email and Anti-Spam (CEAS 2008)*, 2008.

Calais Guerra, P. H., D. Guedes, W. M. Jr., C. Hoepers, M. H. P. C. Chaves ja K. Steding-Jessen, Exploring the spam arms race to characterize spam evolution. *Proc. 7th Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS 2010)*, 2008.

Chang, M.-w., W.-t. Yih ja C. Meek, Partitioned logistic regression for spam filtering. *Proc. 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, sivut 97–105, ACM, New York, NY, USA, 2008.

Conley, K., Symantec June State of Spam Report. Saatavilla: `http://www.symantec.com/connect/blogs/june-state-spam-report`, noudettu 17.5.2011, 2007.

Cook, D., J. Hartnett, K. Manderson ja J. Scanlan, Catching spam before it arrives: domain specific dynamic blacklists. *Proc. Australasian workshops on Grid computing and e-research (AusGrid 2006)*, ACSW Frontiers '06, sivut 193–202, Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 2006.

Cormack, G. V., Email spam filtering: A systematic review. *Found. Trends Inf. Retr.*, *1*(4), sivut 335–455, 2008.

Cormack, G. V. ja A. Kołcz, Spam filter evaluation with imprecise ground truth. *Proc. 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, sivut 604–611, ACM, New York, NY, USA, 2009.

Cormack, G. V. ja T. R. Lynam, TREC 2005 spam track overview. *Proc. 14th Text REtrieval Conference*, 2005.

Cormack, G. V. ja T. R. Lynam, TREC 2007 spam track overview. *Proc. 16th Text REtrieval Conference*, 2007.

Cormack, G. V. ja M. Mojdeh, Autonomous personal filtering improves global spam filter performance. *Proc. 6th Conference on Email and Anti-Spam (CEAS 2009)*, 2009.

Cranor, L. F. ja B. A. LaMacchia, Spam! *Commun. ACM*, *41*(8), sivut 74–83, 1998.

Crocker, D., Challenges in anti-spam efforts. *The Internet Protocol Journal*, *8*(4), sivut 2–14, 2005.

Cunningham, P., N. Nowlan, S. J. Delany ja M. Haahr, A case-based approach to spam filtering that can track concept drift. *Proc. In The ICCBR'03 Workshop on Long-Lived CBR Systems*, 2003.

De Guerre, J., The mechanics of Vipul's Razor technology. *Network Security*, *2007*(9), sivut 15–17, 2007.

Delany, S. J., P. Cunningham, A. Tsymbal ja L. Coyle, A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems*, *18*(4–5), sivut 187–195, 2005.

Duan, Z., Y. Dong ja K. Gopalan, DMTP: Controlling spam through message delivery differentiation. *Comput. Networks*, *51*(10), sivut 2616–2630, 2007.

Eggert, L., DKIM deployment trends. Deployment Trends for IETF Protocols, saatavilla: `https://fit.nokia.com/lars/meter/dkim.html`, noudettu 25.5.2011, 2011a.

Eggert, L., SPF deployment trends. Deployment Trends for IETF Protocols, saatavilla: `https://fit.nokia.com/lars/meter/spf.html`, noudettu 25.5.2011, 2011b.

Felegyhazi, M., C. Kreibich ja V. Paxson, On the potential of proactive domain blacklisting. *Proc. 3rd USENIX conference on Large-scale exploits and emergent threats (LEET'10)*, sivut 6–6, USENIX Association, Berkeley, CA, USA, 2010.

Freed, N. ja N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045 (Draft Standard), updated by RFCs 2184, 2231, 5335, 1996.

FTC, The CAN-SPAM Act: A Compliance Guide for Business. Saatavilla: `http://business.ftc.gov/documents/bus61-can-spam-act-compliance-guide-business`, noudettu 11.5.2011, 2008.

Gomes, L. H., C. Cazita, J. M. Almeida, V. Almeida ja J. Wagner Meira, Workload models of spam and legitimate e-mails. *Performance Evaluation*, *64*(7-8), sivut 690–714, 2007.

Goodman, J. ja W.-t. Yih, Online discriminative spam filter training. *Proc. 3th Conference on Email and Anti-Spam (CEAS 2006)*, 2006.

Görling, S., An overview of the Sender Policy Framework (SPF) as an anti-phishing mechanism. *Internet Research: Electronic Networking Applications and Policy*, *17*(2), sivut 169–179, 2007.

Graham, P., A plan for spam. Saatavilla: `http://www.paulgraham.com/spam.html`, noudettu 5.8.2007, 2002.

Graham, P., Better bayesian filtering. *Proc. MIT Spam Conference*, saatavilla: `http://www.paulgraham.com/better.html`, noudettu 5.8.2007, 2003.

Graham-Cumming, J., Why I hate challenge/response. JGC's Anti-Spam Newsletter, saatavilla: `http://www.jgc.org/writing.html`, noudettu 19.5.2011, 2005.

Guzella, T. S. ja W. M. Caminhas, Review: A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, *36*(7), sivut 10,206–10,222, 2009.

Harris, E., The Next Step in the Spam Control War: Greylisting. Saatavilla: `http://projects.puremagic.com/greylisting/whitepaper.html`, noudettu 24.5.2011, 2003.

Hayati, P. ja V. Potdar, Evaluation of spam detection and prevention frameworks for email and image spam: a state of art. *Proc. 10th International Conference on Information Integration and Web-based Applications & Services*, sivut 520–527, ACM, New York, NY, USA, 2008.

Hayes, B., How Many Ways Can You Spell V1@gra? *American Scientist*, *95*(4), sivut 298–302, 2007.

Jung, J. ja E. Sit, An empirical study of spam traffic and the use of DNS black lists. *Proc. ACM SIGCOMM conference on Internet measurement*, sivut 370–375, ACM Press, New York, NY, USA, 2004.

Junod, J., Servers to spam: drop dead. *Communication News*, *34*(9), sivut 78–79, 1997.

Klensin, J., Taking another look at the spam problem. *The Internet Protocol Journal*, *8*(4), sivut 15–19, 2005.

Klensin, J., Simple Mail Transfer Protocol. RFC 5321 (Draft Standard), 2008.

Kołcz, A. ja A. Chowdhury, Hardening fingerprinting by context. *Proc. 4th Conference on Email and Anti-Spam (CEAS 2007)*, 2007.

Levine, J., DNS Blacklists and Whitelists. RFC 5782 (Informational), 2010.

Levine, J. R., Experiences with greylisting. *Proc. 2nd Conference on Email and Anti-Spam (CEAS 2005)*, 2005.

Li, K. ja Z. Zhong, Fast statistical spam filter by approximate classifications. *Proc. Conf. Measurement and modeling of computer systems*, sivut 347–358, ACM Press, New York, NY, USA, 2006.

Loder, T., M. V. Alstyne ja R. Wash, An economic response to unsolicited communication. *Advances in Economic Analysis & Policy*, *6*(1), sivut 2:1–2:36, 2006.

Louis, G., Bogofilter Calculations: Comparing Bayes Chain Rule with Fisher's Method for Combining Probabilities. Saatavilla: http://www.bgl.nu/bogofilter/BcrFisher.html, noudettu 27.7.2007, 2003.

MAAWG, Managing Port 25 for Residential or Dynamic IP Space. *Recommendation*, The Messaging Anti-Abuse Working Group, San Francisco, CA, USA, saatavilla: http://www.maawg.org/system/files/pubdocs/MAAWG_Port25rec0511.pdf, noudettu 12.5.2011, 2005.

MAAWG, Email Metrics Program: The Network Operators' Perspective – Third and Fourth Quarter 2010. *Report 14*, The Messaging Anti-Abuse Working Group, San Francisco, CA, USA, saatavilla: http://www.maawg.org/sites/maawg/files/news/MAAWG_2010_Q3Q4_Metrics_Report_14.pdf, noudettu 12.5.2011, 2011.

Massey, B., M. Thomure, R. Budrevich ja S. Long, Learning spam: simple techniques for freely-available software. *Proc. USENIX Annual Technical Conference*, sivut 13–13, USENIX Association, Berkeley, CA, USA, 2003.

McAfee, Global S.P.A.M. diaries. July 2008 spam report, saatavilla: http://us.mcafee.com/en-us/local/docs/Spam_Report_July08.pdf, noudettu 13.5.2011, 2008.

McAfee, The carbon footprint of email spam report. Saatavilla: http://resources.mcafee.com/content/NACarbonFootprintSpam, viitattu 18.5.2011, 2009.

MessageLabs Intelligence, Messagelabs Intelligence report. Saatavilla: http://www.messagelabs.com/mlireport/MLIReport_2009.07_July_FINAL.pdf, noudettu 13.5.2011, 2009.

Metsis, V., I. Androutsopoulos ja G. Paliouras, Spam filtering with naive bayes – which naive bayes? *Proc. 3th Conference on Email and Anti-Spam (CEAS 2006)*, 2006.

Mojdeh, M. ja G. V. Cormack, Semi-supervised spam filtering: does it work? *Proc. 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, sivut 745–746, ACM, New York, NY, USA, 2008.

Moore, K., Recommendations for Automatic Responses to Electronic Mail. RFC 3834 (Proposed Standard), updated by RFC 5436, 2004.

Mutchler, A., CAN-SPAM versus the european union e-privacy directive: Does either provide a solution to the problem. *43 Suffolk U. L. Rev.*, *XLIII*(4), sivut 957–981, 2010.

O'Brien, C. ja C. Vogel, Spam filters: bayes vs. chi-squared; letters vs. words. *Proc. 1st international symposium on Information and communication technologies*, sivut 291–296, Trinity College Dublin, 2003.

Pelletier, L., J. Almhana ja V. Choulakian, Adaptive filtering of spam. *Proc. CNSR*, sivut 218–224, IEEE Computer Society, 2004.

Postel, J., Simple Mail Transfer Protocol. RFC 821 (Standard), obsoleted by RFC 2821, 1982.

Puri, R., Bots & botnet: An overview. *Tech. rep.*, SANS Institute, 2003.

Qi, H., X. He, Y. Han, M. Yang ja S. Li, Information theory based feature valuing for logistic regression for spam filtering. *Proc. 2010 International Conference on Asian Language Processing (IALP)*, sivut 166–169, 2010.

Rajab, M. A., J. Zarfoss, F. Monrose ja A. Terzis, A multifaceted approach to understanding the botnet phenomenon. *Proc. ACM SIGCOMM on Internet measurement*, sivut 41–52, ACM Press, New York, NY, USA, 2006.

Robinson, G., A statistical approach to the spam problem. *Linux J.*, *2003*(107), sivut 3, 2003.

Sahami, M., S. Dumais, D. Heckerman ja E. Horvitz, A Bayesian approach to filtering junk e-mail. *Proc. AAAI-98 Workshop on Learning for Text Categorization*, 1998.

Sculley, D., Advances in online learning-based spam filtering, Ph.D. thesis, Tufts University, 2008.

Sculley, D., Re: Rosvm - any real world implementations? Henkilökohtainen sähköposti 11.4.2011, 2011.

Sculley, D. ja G. V. Cormack, Filtering email spam in the presence of noisy user feedback. *Proc. 5th Conference on Email and Anti-Spam (CEAS 2008)*, 2008.

Sculley, D. ja G. V. Cormack, Going mini: Extreme lightweight spam filters. *Proc. 6th Conference on Email and Anti-Spam (CEAS 2009)*, Mountain View, CA, USA, 2009.

Sculley, D. ja G. M. Wachman, Relaxed online SVMs for spam filtering. *Proc. 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, sivut 415–422, ACM, New York, NY, USA, 2007.

Segal, R., Combining global and personal anti-spam filtering. *Proc. 4th Conference on Email and Anti-Spam (CEAS 2007)*, 2007.

Sheng, S., B. Wardman, G. Warner, L. F. Cranor, J. Hong ja C. Zhang, An empirical analysis of phishing blacklists. *Proc. 6th Conference on Email and Anti-Spam (CEAS 2009)*, Mountain View, CA, USA, 2009.

Soma, J., P. Singer ja J. Hurd, Spam still pays: The failure of the CAN-SPAM Act of 2003 and proposed legal solutions. *Harvard Journal on Legislation*, *45*(1), sivut 165–198, 2008.

Spamhaus Project, The definition of spam. Saatavilla: `http://www.spamhaus.org/definition.html`, noudettu 16.5.2011, 2003.

Sullivan, T., The more things change: Volatility and stability in spam features. *Proc. MIT Spam Conference*, saatavilla: `http://www.qaqd.com/research/mit04sum.html`, noudettu 5.8.2007, 2004.

Swartz, N., Deleting spam costs businesses billions. *Information Management Journal*, *May/Jun*, 2005.

Taylor, B., Sender reputation in a large webmail service. *Proc. 3th Conference on Email and Anti-Spam (CEAS 2006)*, 2006.

Taylor, B., D. Fingal ja D. Aberdeen, The war against spam: A report from the front line. *Proc. NIPS 2007 Workshop on Machine Learning in Adversarial Environments for Computer Security*, 2007.

Viestintävirasto, Määräys sähköpostipalvelujen tietoturvasta ja toimivuudesta. 11 A/2008 M, saatavilla: `http://www.ficora.fi/attachments/suomiry/5AWLwAxxQ/Viestintavirasto11A2008M.pdf`, viitattu 20.5.2011, 2008.

Wittel, G. ja S. Wu, On Attacking Statistical Spam Filters. *Proc. 1st Conference on Email and Anti-Spam (CEAS 2004)*, Mountain View, CA, USA, 2004.

Wong, M. ja W. Schlitt, Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. RFC 4408 (Experimental), 2006.

Xu, J. M., G. Fumera, F. Roli ja Z. H. Zhou, Training SpamAssassin with Active Semi-supervised Learning. *Proc. 6th Conference on Email and Anti-Spam (CEAS 2009)*, CEAS, Mountain View, CA, USA, 2009.

Zdziarski, J. A., *Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification*. No Starch Press, San Francisco, CA, USA, 2005.

Zhang, H., The optimality of naive bayes. *Proc. 17th International FLAIRS Conference*, toim. V. Barr ja Z. Markov, AAAI Press, FL, USA, 2004.

Zhang, L., The CAN-SPAM Act: An unsufficient response to the growing spam problem. *Berkeley Tech. L. J.*, *20*(1), sivut 301–332, 2005.

Zheleva, E., A. Kołcz ja L. Getoor, Trusting spam reporters: A reporter-based reputation system for email filtering. *ACM Trans. Inf. Syst.*, *27*(1), sivut 3:1–3:27, 2008.

Zhong, Z., L. Ramaswamy ja K. Li, Alpacas: A large-scale privacy-aware collaborative anti-spam system. *Proc. The 27th Conference on Computer Communications (IEEE INFOCOM 2008)*, sivut 556–564, 2008.