Primena vizuelizacije ansamblova u domenu bezbednosti računarskih mreža

Luka Mladenović,
student osnovnih akademskih studija na
Katedri za računarstvo i informatiku
na Elektronskom fakultetu
Aleksandra Medvedeva 14, 18000 Niš,
Srbija
luka.mladenovic@elfak.rs

***Sadržaj -** Sve više i više korisnika Interneta znači sve veća i veća potreba za povećanom bezbednošću računarskih mreža. Potrebno je što pre uočiti potencijalno opasne i zlonamerne aktivnosti na mreži i imati mehanizam za odbranu od njih. U ovom radu opisan je način detekcije potencijalno opasnog i sumnjivog saobraćaja i aktivnosti na računarskoj mreži Elektronskog fakulteta primenom aglomerativnog hijerarhijskog klusterovanja podataka sa mreže ogranizovanih u ansambl. Za ovo, korišćene su biblioteke za matematičku analizu u programskom jeziku Python, kao NumPy I sckit-learn. Takođe, ove biblioteke pružaju načine vizualizacije prikupljenih podataka, kao u obliku dendograma i tzv. "hot-mapa" koje su se pokazale kao lak i intuitivan način za razumevanje i analizu "trend-ova" u velikoj količini podataka.*

I. Uvod

Svet se sve više i više oslanja na računarske mreže. Zbog učestanosti i uspešnosti napada na računarske mreže, analiza bezbednosnih podataka računarskih mreža i njihova vizualizacija kao ansambl podataka je postala vrlo bitna grana računarske nauke. Kako bi se održala pouzdanost i bezbednost računarske mreže, analitičari konstantno prikupljaju ogromne količine podataka koje obuhvataju najbitnije karakteristike mreža i vrše analize nad datim podacima kako bi uočili napade i sumnjive aktivnosti računara na mreže skrivene u samom saoubraćaju mreže. Vizualizacija ovih podataka se pokazala kao vrlo intuitivan i lak način za istraživanje velikih skupova podataka kao i za njihov prikaz. Cilj ovog rada je projektovanje samo dela pipeline-a za analizu i prikaz podataka jedne računarske mreže. Deo koji je projektovan je sama analiza podataka kao i rudimentaran način njihov prikaza. Naime, cilj je bio analizirati saobraćaj sa mreže Elektronskog fakulteta i otkriti nepravilnosti, sumnjive aktivnosti ili u najgorem slučaju napade na ovu mrežu i prikazati rezultate ove analize na što intuitivniji način. Pored vizuelizacije, druga glavna tema ovog rada je analiza ansambl podataka. U brojnim naučnim disciplinama, istraživači prikupljaju podatke dobijene iz višestrukih iteracija simulacije nekog događaja ili eksperimenta, svaka od kojih ima male promene u početnim parametrima. Kolekcija ovako dobijenih skupova podataka se naziva ansambl i koristi se za simulaciju kompleksnih sistema, istraživanje nepoznatih parametara u inicijalnim uslovima kao i njihov uticaj na ostatak sistema, za smanjivanje nepoznatih uticaja i poređenje strukturnih prikaza različitih modela. Svaki član ovako dobijenog skupa podataka naziva se član ansambla.

Iako na prvi pogled, podaci o jednoj računarskoj mreži i ansambl podataka mogu izgledati kao različiti po strukturi, mogu se smatrati sličnim u polgedu problema koji se mogu pojaviti pri njihovoj analizi i ciljevima analize istih. Oba skupa podataka su veliki i vremenska komponenta podataka igra ulogu u oba skupa. Ovo govori o potrebi analize podataka u vremenskom domenu kao i ostavljanje prostora za skalabilnost tj. proširenje datog skupa podataka. Vizualizacija ansambl podataka se bazira na poređenju i agregaciji članova datog ansambla, dok se vizuelizacija bezbednosih podataka sa mreže fokusira na istraživanju korelacija između različitih tokova saobraćaja na mreži. Ako se podaci sa računarske mreže pretvore u oblik ansambl podataka, moguće je primeniti tehnike za analizu ansamblova kako bi smo poboljšali bezbednosnu analitiku mreža. Kao što je napomenuto, potrebno je prvo organizovati podatke s mreže u ansambl. U ovom radu, ansambl predstavljaju međusobno povezane, vremenski zavisne sekvence alertova, gde svaka sekvenca predstavlja član ansambla. Ostavlja se prostor za definisanje načina povezivanja alerta u jednu sekvencu, birajući prozor vremena u kojem svi dobijeni alerti pripadaju istoj sekvenci. Nakon prikupljanja podataka, računamo „razlike" (eng. Dissimilarity) između članova ansambla korišćenjem tehnika za poređenje vremenski osetljivih podataka u vremenskom domenu. Dobijene razlike između članova koristimo kako bi smo primenili aglomerativno hijerarhijsko klasterovanje i grupisali članove koji su najsličniji. Ovakav prikaz prikupljenih podataka povećava šanse da analitičar mrežnog saobraćaja na vreme prepozna i otkloni potencijalne pretnje i sumnjive aktivnosti u datoj mreži.

U narednom poglavlju, data je teorijska osnova potrebna za implementaciju i razumevanje sistema navedenog u ovom radu. Biće objašnjenja o načinu kreiranje ansambl podataka iz bezbednosnih podataka sa mreže, načinu poređenja različitih članova ansambla kao i načinu vizuelizacije dobijenih rezultata. U trećem poglavlju, biće opisana implementacija predloženog sistema. Četvrto poglavlje biće posvećeno prikazu rada datog sistema. U petom poglavlju biće dat zaključak.

II. TEORIJSKA OSNOVA

II.A.Analiza ansambl podataka

U naučnim domenima, ansambl je definisan kao skup međusobno povezanih skupova podataka, gde svaki formira član ansambla. Svaki član sadrži podatke prikupljene u različitim vremenskim trenucima za iste prozore vremena.

Istraživanje članova ansambla i njihovo međusobno poređenje su dva najvažnija zadatka analitičara. Kako bi smo primenili tehnike vizuelizacije ansambla na bezbednosne podatke sa mreže, potrebno je organizovati te podatke kao ansambl, u zavisnosti od potreba i cilja analize i samog poređenja. Kada je u pitanju analiza alert podatka sa mreže, skup podataka se sastoji iz izvorne i odredišne IP adrese, porta, vremenskih podataka (start, end, duration), protokola, poruke i klasifikacije.

Kako bismo analizirali odnose između članova alert ansambla, potrebno je prvenstveno definisati šta će biti član ansambla. Kada je reč o bezbednosnim podacima mreže, član mogu biti sve kombinacije odredišna IP adresa/port koji komuniciraju sa jednom datom izvorišnom IP adresom. Nakon toga, potrebno je date alerte izdvojene po članovima dalje urediti po tzv. vremenskim koracima. Vremenski domen jednog ansambl člana je podeljen, uniformno, na vremenske korake tj. manje vremenske domene. Naime, počev od najranijeg alerta u datom članu, potrebno je združiti i izbrojati alerte u jednom vremenskom koraku. Poređenje ansambl članova se vrši poređenjem promena u broju alertova po vremenu.

II.B.Poređenje članova ansambla

Prednost korišćenja tehnika za vizuelizaciju ansamblova je mogućnost fokusiranja na odnose između samih članova ansambla. Ovo je najčešće bitno i korisno u situacijama gde pokušavamo da identifikujemo povezane članove ili one čija je aktivnost ili promena neuobičajena (kao na primer bezbednosti podaci jedne mreže). Kako bismo prikazali prirodu odnosa između članova ansambla i utvrdili koliko se međusobno razlikuju, potrebno je kreirati matricu razlika (eng. Dissimilarity matrix). Za članove ansambla koji su poravnati po vremenskoj osi, možemo iskoristiti Menheten metriku za računanje razlike između njih. Za one koji nisu poravnati po vremenskoj osi, potrebno je koristiti tehniku pozantu kao DTW(eng. Dynamic Time Warping). Uz pomoć ove tehnike, moguće je naći optimalno poklapanje između članova koji nisu poravnati po vremenskoj osi.

Član ansambla m_i je sekvenca združenih podataka (tj. broja alertova) sakupljenih u t vremenskih koraka koje analitičar može odabrati. $mi = (n_{i,1}, n_{i,2} \ldots, n_{i,t})$.

Treba imati u vidu da m ovde predstavlja član ansambla tj. niz vrednosti n, gde $n_{i,k}$ predstavlja broj alertova sakupljenih za *i-ti* član ansambla u *k-tom* vremenskom koraku. M ovde predstavlja jedan par odredišne IP adresa I porta uparenih sa konkretnom izvorišnom IP adresom, a n je broj alerta zabeleženih u datom vremenskom koraku.

Ako su članovi ansambla poravnati po vremenskoj osi, može ih relativno lako međusobno uporediti korišćenjem Menheten metrike za definisanje razlike. Ako je $dis_{i,j}$ element matrice razlika, on predstavlja razliku između članova m_i i m_j. Menheten razlika se računa kao:

$$dis_{i,j} = \Sigma \mid n_{i,p} - n_{j,p} \mid / t,\ p = 1,t.$$

Nažalost, mnogi članovi ansambla podataka sakupljenih iz realnog sveta se ne poklapaju po vremenskoj osi i zato je potrebno primeniti DTW. Ideja je kreirati „1 na više“ i „više na 1“ relacija između elemenata sekvenci koje upoređujemo kako bi uspeli da minizujemo ukupnu distancu tj. razliku između sekvenci. Najčešće se koristi u audio analizi, gde možemo porediti zvuke sa talasima istog oblika ali sa različitim vremenima početka. Na grafikonu gde bi recimo y osa bila amplituda, a x vremene, ako koristimo Menheten distancu, ova dva talasa, iako identična, ne bi bila prepoznata kao takva. DTW nam omogućava da i ovakve situacije tačno protumačimo.

Analogija sa našim problem bi bila da se promene u alert saobraćaju (tj. broj alerta) menja kroz vreme i to u različitim vremenskim koracima. Kako bi smo ove skupove mogli da poredimo i nađemo potencijalne sličnosti i razlike, koritstimo DTW. Pokušavamo da pronađemo idealni način poređenja između dve vremenske sekvence koje se mogu razlikovati po brzini i frekvenciji promena. DTW algoritam nalazi nelinearan način poređenja elemenata takav da minimizira ukupnu razdaljinu (razliku) između članova sekvenci. Kako bi pronašao najmanju udaljenost između elemenata m_i i m_j, DTW algoritam kreira matricu W dimenzija $t * t$, gde je t broj vremenskih koraka na koje smo podelili vremenski domen sekvence. Element W(u,v) predstavlja razliku između u-tog i v-tog vremenskog koraka elemenata m_i i m_j,. Dinamičkim programiranjem, dolazi se do najmanje razlike između elemenata m_i i m_j na osnovu matrice W. DTW algoritam je sporiji i kompleksniji nego prosto računanje razlike Menheten metrikom, ali nam omogućava tačnu analizu drugačijih sekvenci.

Ako je $D'(m_{i,u}, m_{j,v})$, $1 \le u,v \le t$, najmanja distanca između sekvenci $m_{i,u} = (n_{i,1}, n_{i,2}, \ldots, n_{i,u})$ i $m_{i,j} = (n_{i,1}, n_{i,2}, \ldots, n_{i,j})$. Vrednost DTW razlike $dis_{i,j} = D'/t$ dobija se uspomoć dinamičkog programiranja sledećom rekurzijom:

$$D'(m_{i,u}, m_{j,v}) = dis(n_{i,u}, n_{j,v}) + min(D'(m_{i,u-1}, m_{j,v}), D'(m_{i,u-1}, m_{j,v-1}), D'(m_{i,u}, m_{j,v-1}))$$

gde je

$$dis(n_{i,u}, n_{j,v}) = \mid n_{i,u} - n_{j,v} \mid$$

C. Aglomerativno klasterovanje

Aglomerativno klasterovanje je algoritam koji radi po „bottom-up“ principu. Svaki od objekata koji razmatramo je prvobitno dodeljen svom klasteru, tako da na najnižem nivou stabla imamo onoliko klastera koji imamo objekata koje pokušavamo da klasterizujemo. U svakom koraku algoritma, dva klastera koja su najsličnija su uparuju u jedan novi klaster. Ova procedura se ponavlja sve dok ne dobijemo jedan jedini klaster koji sadrži sve elemente. Pored ovakvog pristupa, postoje i tzv. DIANA (Divisive Analysis) algoritmi koji su invers aglomerativnog klasterovanja; počinjemo od jednog klastera u kojem se nalaze svi objekti i u svakom koraku odvajamo najheterogeniji klaster u dva nova., da bismo na kraju dobili onoliko klastera koliko imamo objekata. Aglomerativno hijerarhijsko klasterovanje je pogodno kada klasteri koje pokušavamo da nađemo nisu preveliki u odnosu na ukupan broj objekata. Pre nego što možemo da primenimo ovaj algoritam, potrebno je da pripremimo podatke. Koraci potrebni za pripremu su opisani u prethodna dva podpoglavlja i to su organizacija podataka u oblik pogodan za klasterovanje, npr. ansambl kao i računanje razlika (distanci) između elemenata koje želimo da klasterizujemo. Naredni korak se naziva „povezanost“ (eng.Linkage) i tiče se različitih načina na koje možemo da defnišemo razliku između klastera koje se javljaju tokom

izvršenja. U bibliotekama koje implementiraju aglomerativno klasterovanje postoji mogućnost biranja načina na koji će algoritam računati razlike između klastera i po kojim kriterijumima će dva klastera spojiti u jedan.

Najpoznatiji načini klasterovanja su:

- Maksimalna ili kompletna povezanost: Razlika između dva klastera se definiše kao maksimalna vrednost između svakih od parova razlika svih elemenata iz klastera 1 i klastera 2. Ovakva povezanost daje „zbijenije“ klastere.
- Minimalna ili jedinstvena povezanost: Razlika između dva klastera se definiše kao minimum razlike svakih od parova elemanata iz klastera 1 i klastera 2. Ovakva povezanost daje duže i „labavije“ klastere.
- Srednja ili prosečna povezanost: Razlika između dva klastera se definiše kao srednja vrednost vrednosti razlike elemenata iz klastera 1 i klastera 2.
- Centralna povezanost: Razlika između dva klastera se definiše kao razlika između centroida klastera 1 i centroida klastera 2. Centroid je objekat u datom klasteru koji se tretira kao centar klastera. Obično se nalazi podjednako udaljeno od svih ivica klastera.
- Ward-ova metoda minimalne varijanse: Teži da minimizira varijansu unutar samog klastera. Koristi se sa nekom od prethodno pomenutih metoda kako bi uparila dva klastera koja smatra najsličnijim. Najčešće se koristi minimalna povezanost.

Dendrogrami će takođe biti korišćeni za vizuelizaciju podataka dobijenih na kraju analize, naime za prikaz samih klastera kojima članovi ansambla pripadaju. Dendrogrami se za to inače i koriste; za grafički prikaz hijerarhije klastera.

III. Pregled implementiranog sistema

Sistema predložen u ovom radu je relativno jednostavan u pogledu arhitekture samog sistema. Za analizu podataka korišćen je Python i open-source biblioteke koje su dostupne u Python-u za primenu algoritama poput DTW i aglomerativnog klasterovanja, kao npr. NumPy, Pandas i sckit-learn. Za prikaz podataka je takođe korišćena biblioteka pod nazivom matplotlib.

Podatke potrebne za izradu ovog rada dobijeni su od prof. dr. Vladimir Ćirić. Potencijalno proširenje ovog rada bi bilo projektovanje kompletnog pipeline-a, gde bi se podaci preuzimali sa ElasticSearch-a, obrađivali u programu opisanom u ovom projektu i slali na vizuelizaciju preko alata pod nazivom Kibana.

Dobijene podatke je prvenstveno trebalo „očistiti“ tj. organizovati ih tako da za svaki par odredišna IP adresa, port imamo sve podatke potrebne za dalju analizu. Svaki od unosa sa istom kombinacijom odredišna IP adresa, port bi predstavljao jedan alert za datu komunikaciju. U radu je izabrana analiza članova ansambla dobijenih za jednu konkretnu IP adresu, potencijalna proširenja ovog rada mogu analizirati izgled dendrograma i „hot-mapa“ za svaku moguću izvorišnu IP adresu na mreži.

Zatim, radi lakšeg rada, vremena su sortirana, počev od najskorijeg za svaki od ansambl članova koji smo dobili u prethodnom koraku. Ovo je urađeno kako bi naredni korak u analizi bio olakšan, a to je deljenje vremenskog domena datog ansambl člana na manje vremenske korake i prebrojvanje alerta u svakom od njih.

Počev od prvog člana, potrebno je proći kroz sekvencu alertova i u datim vremenskim koracima prebrojati koliko se alerta javilo. Alert bi pripadao jednom vremenskom koraku ako je vreme njegove pojave između početka i kraja datog vremenskog koraka. Na ovaj način, možemo pratiti promenu po broju alertova za datu komunikaciju na mreži i uporediti je sa ostalima, kako bi smo analizirali ponašanje svakog od računara na mreži.

Nakon pripreme podataka tj. kreiranje ansambla, možemo izračunati matricu razlika. Za svaki par članova u ansamblu poziva se prethodnopomenuti DTW algoritam, koji će bez obzira na potencijalnu neusklađenost u vremenskom domenu naći optimalan način za poređenje dva člana i vratiti tačnu vrednost razlike između dva data člana ansambla. Svaku dobijenu vrednost upisujemo u matricu *M* koju ćemo kasnije prikazati. Vrednost *M(i,j)* predstavlja razliku između *i-tog* i *j-tog* člana ansambla. U našem primeru, matrica *M* bi bila dimenzija 50 × 50.

IV. Prikaz dobijenih rezultata

Na slici 4.0 prikazani su podaci korišćeni tokom izrade ovog rada. Sa slike, jasno se može uočiti koja polja opisuju odredišnu IP adresu kao i port. Takođe “start” polje nam opisuje vreme početka događaja tj. alerta a polje “duration” trajanje samog alerta.

Za prikaz matrice M dobijene kao rezultate obrade datih podataka korišćena je biblioteka matplotlib. U okviru nje je bilo moguće definisati način mapiranja boja za prikaz date matrice. U ovom radu odabran je crno-beli prikaz tj. crna polja u matrici ukazuju na veliku razliku između datih članova ansambla dok bela boja ukazuje na vrlo malu razliku. Najveći deo matrice je u nijansama sive boje, što bi ukazivalo da iako se ovi članovi razlikuju, razlika nije drastična ili je barem nedovoljna da ukaže na neke ozbiljnije probleme ili napade.

Takođe, matricu M možemo iskoristiti za prikaz dendrograma datih članova ansambla, gde metode pružene kroz bibliotečki API omogućavaju da se odabere način povezanosti tj. način na koji će algoritam aglomerativnog hijerarhijskog klasterovanja da nalazi i tretira razlike između klastera. U ovom radu, odabrana je „Ward-ova metoda minimalne varijanse“ kako bi klasteri dobijeni bili ispunjeni što je više moguće homogenim elementima. Postoji prostor za dalji rad ovde u pogledu drugačijih parametara za povezanost klastera i diskusije o potencijalnim prednostima drugačijih pristupa.

Na slikama 4.1. i 4.2. prikazani su rezultati dobijeni u toku ovog rada. Na slici 4.2., po rasporedu boja u datoj matrici, može videti da su drastične razlike retke između članova ansambla. Ovi rezultati su pokazatelj da je reč o mreži bez mnogo promena po broju alertova u jedinici vremena, gde je sumnjiva aktivnosti umreženih računara relativno retka.

Takođe, na slici 4.2 možemo videti rezultate aglomerativnog klasterovanja na dati ansambl podataka. Vidimo da su inicijalno svih 50 članova ansambla u zasebnim klasterima i kako algoritam nastavlja s radom, tako se po dva najsličnija klastera spajaju u jedan, da bi smo na kraju dobili jedan klaster

sa svim članovima ansambla u njemu. Sudeći po izgledu dendrograma, članovi ansambla se mogu grupisati u 5 ili 6 većih klastera, svaki sa svojim posebnim karakteristikama. Ovo je takođe prostor potencijalnih daljih radova, dalja analiza dobijenih klastera radi otkrivanja skrivenih sličnosti ili razlika između njih koje mogu biit od pomoći za povećanje bezbednosti računarskih mreža.

V. Zaključak

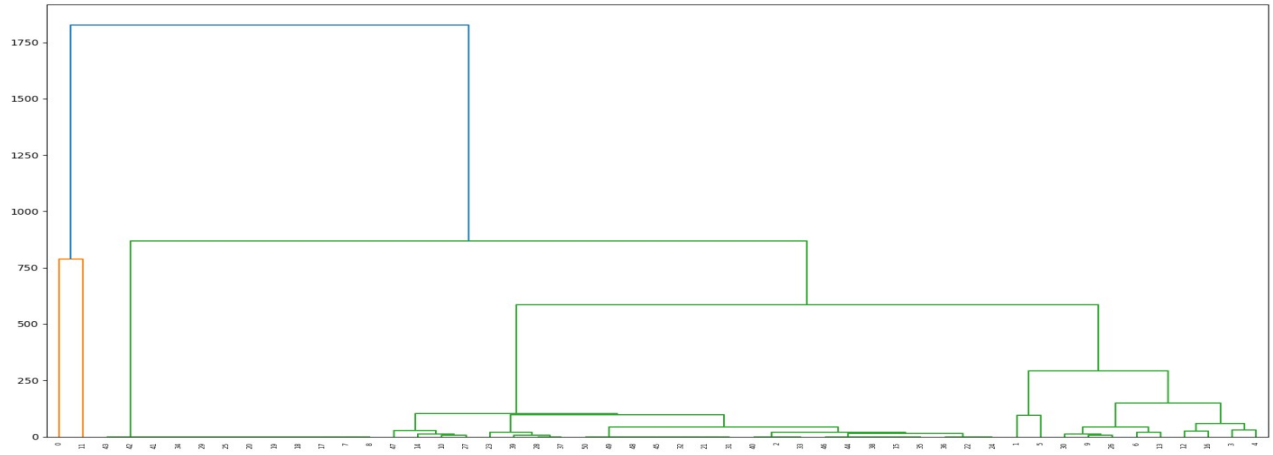
U ovom radu je implementiran sistem primene aglomerativnog hijerarhijskog klasterovanja na ansambl bezbednosnih podataka jedne računarske mreže kao i način vizuelizacije dobijenih podataka uz pomoć tehnika za analizu ansambl podataka.

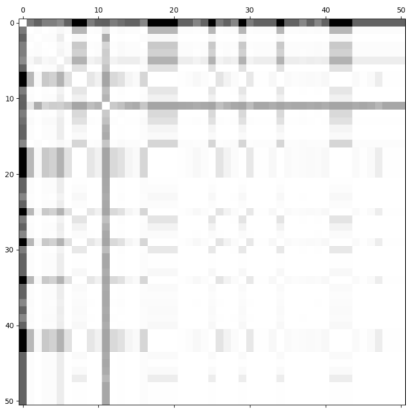

```
{"type":"http",
"status":"OK","source":{"port":36986,"bytes":557,"ip":"37.35.66.92"},
"destination":{"port":80,"bytes":117603,"domain":"tempus.elfak.ni.ac.rs","ip":"160.99.13.144"},
"event":{"duration":39417383,"type":["connection","protocol"],"dataset":"http",
"start":"2021-11-21T22:28:41.492Z","end":"2021-11-21T22:28:41.531Z",
"category":["network_traffic","network"],"kind":"event"},


"type":"dns",
"source":{"port":37917,"bytes":37,"ip":"160.99.13.144"},
"destination":{"port":53,"bytes":74,"ip":"160.99.12.230"},
,"server":{"port":53,"bytes":74,"ip":"160.99.12.230"},
"event":{"duration":47068052,"type":["connection","protocol"],
"dataset":"dns","start":"2021-11-21T22:37:16.660Z",
"end":"2021-11-21T22:37:16.708Z","category":["network_traffic","network"],"kind":"event"}
```


Slika 4.0 Prikaz podataka za rad

VI. Zahvalnica

Autor se zahvaljuje prof. dr. Vladimiru Ćiriću i prof. Nađi Gavrilović na uloženom trudu, savetima i pomoći pruženoj tokom izrade ovog rada.





Slika 4.1 i 4.2. Izgled dendrograma (gore) i dissimilarity matrice za 50 članova ansambla (dole)

[1] *O. Alabi, X. Wu, J. Harter, M. Phadke, L. Pinto, H. Petersen, S. Bass, M. Keifer, S. Zhong, C. G. Healey, and R. M. Taylor II. Comparative visualization of ensembles using ensemble surface slicing. In Visualization and Data Analytics, volume 8294, pages 0U: 1–12, San Francisco, CA, 2012.*

[2] P. Senin. Dynamic time warping algorithm review. Technical report, Department of Computer Science, University of Hawai‘i at Manoa, Honolulu, HI, 2008.

[3] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In KDD-94: AAAI-94 Workshop on Knowledge Discovery in Databases, pages 359–370, Seattle, WA, 1994.

[4] T. Taylor, D. Paterson, J. Glanfield, G. C., S. Brooks, and J. McHugh. FlowVis: Flow visualization system. In Proceedings of the Cybersecurity Applications & Technology Conference For Homeland Security 2009 (CATCH ’09), pages 186–198, Washington, DC, 2009.

[5] A. Wilson and K. Potter. Toward visual analysis of ensemble data sets. In Proceedings of the 2009 Workshop on Ultrascale Visualization (UltraVis ’09), pages 48–53, Portland, OR, 2009.

[6] Y. Zhang, Y. Xiao, M. Chen, J. Zhang, and H. Deng. A survey of security visualization for computer network logs. Security and Communication Networks, 5(4):404–421, 2012

[7] Z. Kan, C. Hu, Z. Wang, G. Wang, and X. Huang. NetVis: A network security management visualization tool based on treemap. In Proceedings of the 2nd International Conference on Advanced Computer Control (ICACC 2010), pages 18–21, Shenyang, China, 2010.

[8] P. Kothur, M. Sips, H. Dobslaw, and D. Dransch. Visual analytics for ¨ comparison of ocean model output with reference data: Detecting and analyzing geophysical processes using clustering ensembles. IEEE Transactions on Visualization and Computer Graphics, 19:1893–1902, 2013.