

UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI FAKULTET

Goran M. Rakočević

MAŠINSKO UČENJE U BEŽIČNIM
SENZORSKIM MREŽAMA

doktorska disertacija

Beograd, 2014

UNIVERSITY OF BELGRADE
SCHOOL OF ELECTRICAL ENGINEERING

Goran M. Rakočević

**MACHINE LEARNING IN WIRELESS
SENSOR NETWORKS**

doctoral dissertation

Belgrade, 2014

Mentor:

Prof. Dr Veljko Milutinović, redovni profesor,
Elektrotehnički Fakultet, Univerzitet u Beogradu

Članovi komisije:

Prof. Dr Miroslav Bojović, vanredni profesor,
Elektrotehnički Fakultet, Univerzitet u Beogradu

Prof. Dr Miloš Kovečević, vanredni profesor,
Građevinski Fakultet, Univerzitet u Beogradu

Prof. Dr Dragan Milićev, vanredni profesor,
Elektrotehnički Fakultet, Univerzitet u Beogradu

Datum odbrane: _____

MAŠINSKO UČENJE U BEŽIČNIM SENZORSKIM MREŽAMA

Rezime

Bežične senzorske mreže predstavljaju sisteme koji se sastoje iz većeg broja računarskih uređaja malih dimenzija, koji su opremljeni izvesnim brojem senzora za prikupljanje podataka iz okruženje i koji su sposobni da komuniciraju pomoću radio veze. Bežične senzorske mreže mogu biti izvor velikih količina podataka, pa su pogodne za upotrebu sa tehnikama mašinskog učenja. Cilj ovog rada je prilagođavanje odabranih algoritama mašinskog učenja, tako da se izračunavanja izvršavaju distribuirano, na samim uređajima koji sačinjavaju senzorsku mrežu.

Kroz analizu problema, dobijena su tri osnovna zahteva koje algoritam distribuiranog mašinskog učenja treba da zadovolji pri radu u bežičnoj senzorskoj mreži. Prvo, neophodno je minimizovati komunikaciju između čvorova u mreži, pošto se prilikom komunikacije troši znatna količina energije. Drugo, potrebno je da su rezultati rada algoritma precizni. Treće, potrebno je, u što većoj meri, izbeći upotrebu centralizovanih računarskih resursa, koji su izvan bežične senzorske mreže.

Centralni doprinos ovog rada predstavljaju dva algoritma distribuiranog mašinskog učenja koji su prilagođeni upotrebi u bežičnim senzorskim mrežama. Prvi algoritam je klasifikacioni sistem kod kojega se klasifikacija vrši na dva nivoa. Prvo se na samim čvorovima, primenom proizvoljnog (dat je primer sa stabilna odlučivanja) klasifikatora donosi lokalna odluka. Potom se odluka na nivou mreže donosi težinskom većinskom logikom na osnovu svih lokalnih odluka. Pokazuje se da upotrebom Kapa statistike moguće ostvariti visok nivo tačnosti pri radu ovog algoritma.

Drugi predstavljeni algoritam je algoritam za grupisanje podataka, na osnovu Kohonenovih neuralnih mreža. Arhitektura predložene mreže je prilagođena podacima koji se dobijaju sa senzora u bežičnim senzorskim mrežama. Takvi podaci sadrže u sebi dve informacije, gde se neki događaj desio i tip događaja desio. Na osnovu ovog zapažanja konstruisana je Kohonenova neuralna mreža sa dva sloja, prvi koji služi da odredi lokaciju događaja i drugi koji služi da odredi njegov tip.

Oba algoritma su ispitana matematičkom analizom i pokazano je da nude značajne mogućnosti za energetska uštedu pri radu. Tačnost algoritama je analizirana primenom računarskih simulacija. Zaključuje se da opisani algoritmi mogu imati praktičnu primenu, te da tehnike distribuiranog mašinskog učenja predstavljaju moguće rešenja za održiv razvoj velikih senzorskih mreža.

Ključne reči: mašinsko učenje, bežične senzorske mreže, distribuirana klasifikacija, distribuirani sistemi

Naučna oblast: Elektrotehnika

Uža naučna oblast: Računarska tehnika i informatika

UDK broj: 004.4

Abstract

Wireless sensor networks are systems that consist of a number of small computational devices, called sensor nodes. A sensor node is also equipped with one or more sensors for gathering data from the environment, and is capable of establishing radio communications with other nodes in the network. Wireless sensor networks can produce large amounts of data, making them suitable for applying machine learning techniques. The goal of the work presented here is the adaptation of selected data mining algorithms for distributed execution, so that they can be run directly on the sensor nodes.

Through the analyses of the problem, three basic requirements for a distributed data-mining algorithm suitable for execution in a wireless sensor network have been reached. First, minimizing the communication between the nodes is of paramount importance, since communication consumes significant amounts of energy. Second, the algorithm should yield accurate results. Third, use of centralized computational resources should be avoided as much as possible.

The major contribution of the work presented in this thesis comes in the form of two novel algorithms for distributed machine learning, that are adapted to suit the needs of the execution in a wireless sensor network. First of the two algorithms is a classification scheme that consists out of two layers. The first layer is executed on the sensor nodes, and can utilize any type of a classifier (the approach was tested using J48 decision trees). The results of this step are then propagated to a centralized, second layer classifier that is based on a weighted voting scheme. The second of the two algorithms is a clustering algorithm, based on Kohonen neural networks. The architecture of the network is adapted to the nature of the data that are obtained from sensor nodes in wireless sensor networks. Such data contain two sources of variability: the information about the location of an event that has taken place and is being sensed and the properties of the event that has taken place. Based on this observation, the proposed Kohonen neural network is constructed to contain two layers, the first layer clustering data with respect to the location of origin, and the second layer that clusters data with respect to the underlying event type.

Calculations show that both of the proposed algorithms offer significant savings in terms of energy consumption. The accuracy of the algorithms was verified using computational simulations. In conclusion, the proposed algorithms are suitable for a number of practical applications, while the machine learning techniques present a viable approach to building sustainable large wireless sensor networks.

Key words: machine learning, wireless sensor networks, distributed classification, distributed systems

Naučna oblast: Electrical Engineering

Uža naučna oblast: Computer engineering and information theory

UDK No: 004.4

SADRŽAJ

1. UVOD	3
STRUKTURA TEKSTA KOJI SLEDI.....	4
2. PREGLED STANJA U NAUČNOJ OBLASTI	6
2.1. KATEGORIZACIJA TEHNIKA MAŠINSKOG UČENJA	7
2.2. DISTRIBUCIJA PODATAKA	9
2.3. POSTOJEĆA REŠENJA ZA MAŠINSKO UČENJE U SENZORSKIM MREŽAMA	16
2.3.1. POSTOJEĆA REŠENJA ZA KLASIFIKACIJU.....	17
3.2.1.1 <i>Klasifikacija sa vertikalno distribuiranim podacima.....</i>	<i>17</i>
3.2.1.2 <i>Klasifikacija sa horizontalno distribuiranim podacima.....</i>	<i>21</i>
2.3.2. POSTOJEĆA REŠENJA ZA REGRESIJU.....	24
2.3.2.1. <i>Regresija sa vertikalno distribuiranim podacima.....</i>	<i>25</i>
2.3.2.2. <i>Regresija sa horizontalno distribuiranim podacima.....</i>	<i>27</i>
2.3.3. POSTOJEĆA REŠENJA ZA GRUPISANJE.....	32
2.3.3.1. <i>Grupisanje sa vertikalno distribuiranim podacima.....</i>	<i>32</i>
2.3.3.2. <i>Grupisanje sa horizontalno distribuiranim podacima.....</i>	<i>34</i>
2.3.4. POSTOJEĆA REŠENJA ZA PRONALAZENJE ASOCIJATIVNIH PRAVILA	36
2.3.4.1. <i>Pronalaženje asocijativnih pravila sa vertikalno distribuiranim podacima</i>	<i>36</i>
2.3.4.2. <i>Pronalaženja asocijativnih pravila sa horizontalno distribuiranim podacima.....</i>	<i>39</i>
2.4. DISKUSIJA POSTOJEĆIH REŠENJA.....	41
2.4.1. RAD SA MOBILNIM SENZORIMA I PROMENAMA U OKRUŽENJU	41
2.4.2. VREDNOSTI KOJE NEDOSTAJU	43
2.4.3. OČUVANJE PRIVATNOSTI.....	45
2.4.4. ZAHTEVNOST U POGLEDU KOMUNIKACIJE	46
2.4.5. KAŠNJENJA PRI RADU SISTEMA.....	49
2.4.6. ZAKLJUČCI NA OSNOVU ANALIZE POSTOJEĆIH REŠENJA	52
2.4.7. PERFORMANSE ALGORITAMA ZA DISTRIBUIRANO MAŠINSKO UČENJE U BEŽIČNIM SENZORSKIM MREŽAMA.....	56
3. VERTIKALNO DISTRIBUIRANO MAŠINSKO UČENJE I NOVE TEHNIKE ZA ODREĐIVANJE TEŽINSKIH FAKTORA	59
3.1. SPECIFIKACIJA PROBLEMA	61
3.2. OPIS PREDLOŽENOG REŠENJA.....	63
3.2.1. <i>Scenario predviđene primene senzorske mreže za koju se razvija algoritam distribuiranog mašinskog učenja.....</i>	<i>63</i>
3.2.2. <i>Uslovi i pretpostavke.....</i>	<i>64</i>
3.2.3. <i>Algoritam rada sistema</i>	<i>65</i>
3.2.4. <i>Određivanje vrednosti glasova pojedinih čvorova.....</i>	<i>65</i>
3.2.5. <i>Težinsko glasanje na osnovu kapa-statistike.....</i>	<i>67</i>
3.2.6. <i>Koraci pri izvršavanju algoritma</i>	<i>67</i>
3.2.7. <i>Pravljenje lokalnih modela koji uključuju podatke sa većeg broja čvorova kada među podacima postoje značajne interakcije</i>	<i>69</i>

3.3. ANALIZA RADA SISTEMA NA OSNOVU SIMULACIJE SA SKUPOVIMA PODATAKA.....	71
3.3.1. Metodologija.....	71
3.3.2. Podaci koji su korišćeni prilikom simulacija.....	71
3.3.3. Rezultati simulacija i njihova analiza.....	73
3.3. ANALIZA KOMUNIKACIJE U MREŽI.....	75
3.3.1. Metodologija.....	75
3.3.2. Rezultati analize komunikacije.....	76
3.4. POTROŠNJA ENERGIJE U TOKU RADA SISTEMA.....	77
3.5. ANALIZA RADA SISTEMA KROZ SIMULACIJU REALNOG OKRUŽENJA	81
3.5.1. Metodologija.....	81
3.6. DISKUSIJA PREDLOŽENOG REŠENJA ZA VERTIKALNO DISTRIBUIRANU KLASIFIKACIJU U BEŽIČNIM SENZORSKIM MREŽAMA	85
4. DISTRIBUIRANE KOHONENOVE SAMOORGANIZUJUĆE MAPE U BEŽIČNIM SENZORSKIM MREŽAMA.....	87
4.1. KOHONENOVE NEURALNE MREŽE	87
4.2. TRENIRANJE KOHONENOVE NEURALNE MREŽE	89
4.3. PRIMENA KOHONENOVIH SAMOORGANIZUJUĆIH MAPA U BEŽIČNIM SENZORSKIM MREŽAMA	90
4.4. OPIS PREDLOŽENOG REŠENJA.....	91
4.4.1 Osnovni pristup za upotrebu Kohonenove samoorganizujuće mape u bežičnoj senzorskoj mreži.....	92
4.4.2. Prošireni pristup za upotrebu Kohonenove samoorganizujuće mape u bežičnoj senzorskoj mreži.....	95
4.5. ANALIZA RADA OSNOVNE VERZIJE SISTEMA NA OSNOVU SIMULACIJE	98
4.5.1. Metodologija.....	98
4.6. ANALIZA RADA PROŠIRENE VERZIJE SISTEMA NA OSNOVU SIMULACIJE	101
4.7. DISKUSIJA PREDLOŽENIH REŠENJA ZA UPOTREBU KOHONENOVE SAMOORGANIZUJUĆE MAPE U BEŽIČNOJ SENZORSKOJ MREŽI.....	104
5. ZAKLJUČAK	107
REFERENCE:	111
BIOGRAFIJA AUTORA	116

1. Uvod

Bežične senzorske mreže su sistemi koji se sastoje od malih uređaja koji se nazivaju senzorski čvorovi. Svaki senzorski čvor je opremljen jednim ili većim brojem senzora, centralnim procesorom, kao i sistemom za bežičnu komunikaciju. U tipičnom scenariju upotrebe bežične senzorske mreže senzorski čvorovi se postavljaju u nekoj sredini (na primer u polju ili šumi, ili pak unutar fabrike ili zgrade) gde prikupljaju informacije kroz svoje senzore. Ovi čvorovi formiraju mrežu, posredstvom bežične komunikacije, i razmenjuju podatke dobijene sa senzora, kao i druge informacije (na primer o lokacijama, međusobnim rastojanjima, podatke vezane za sinhronizaciju, itd.). Informacije prikupljene sa senzora se kroz ovako formiranu mrežu transportuju do tzv. izlaznih čvorova (*gateway nodes*) gde napuštaju bežičnu senzorsku mrežu. Izlazni čvorovi su najčešće povezani za računarski server, ili direktno na internet, čime podaci iz senzorske mreže postaju dostupni za dalju obradu i korišćenje.

Praktično ostvarljiva veličina jedne senzorske mreže u znatnoj meri varira sa njenom svrhom, i povećava se sa razvojem tehnologije. Veće mreže danas broje nekoliko stotina do nekoliko hiljada čvorova, uz predviđanja da će u budućnosti brojati i po nekoliko miliona. Kao posledica ovakvog rasta javlja se problem obrade velikih količina podataka koje nastaju u bežičnim senzorskim mrežama. Takođe, treba uzeti u obzir činjenicu da se senzorski čvorovi uglavnom napajaju iz baterija, pri čemu se u radu jedne mreže najveći deo energije potroši u toku komunikacije između čvorova. Pošto veći broj čvorova znači više komunikacije između njih, rast cele mreže ima negativne posledice po potrošnju individualnih čvorova u mreži (posebno važi za čvorove koji se nalaze blizu izlaznog čvora i prenose podatke do njega). Stoga je jasno da je minimizacija komunikacije jedan od najvažnijih izazova u razvoju bežičnih senzorskih mreža.

Mogući pristup koji vodi efikasnom iskorišćenju bežične senzorske mreže i podataka koji nastaju u njoj bazira se na upotrebi tehnika mašinskog učenja. Mašinsko učenje je skup pristupa analizi i obradi (potencijalno veoma velikih količina) podataka u potrazi za obrascima i pravilnostima koji bi omogućili pronalaženje nekih informacija ili znanja sadržanih u datim podacima. S obzirom na to da se često koriste sa velikim količinama podataka, tehnike mašinskog učenja su pogodne za rad sa velikim senzorskim mrežama, i donekle rešavaju problem obrade velike količine nastalih podataka.

U klasičnom pristupu upotrebe mašinskog učenja rad sa podacima iz senzorskih mreža uključuje dva koraka. Prvo se svi podaci iz mreže prikupljaju i skladište na nekoj centralizovanoj lokaciji, potom se ovako prikupljeni podaci analiziraju. Opisani pristup ima dva bitna nedostatka: zahteva visok nivo komunikacije u bežičnoj senzorskoj mreži kako bi se svi podaci skupili na jednom mestu, i računarski kapacitet na senzorskim čvorovima se ne koristi.

Ova teza se bavi pristupom koji ide jedan korak dalje: analizom i razvojem algoritama *distribuiranog* mašinskog učenja, pri čemu se sami algoritmi, ili njihovi delovi, izvršavaju unutar bežične senzorske mreže, pre nego što su podaci preneti na jednu centralizovanu lokaciju. Na ovaj način se umanjuje količina podataka koju je potrebno prenositi kroz mrežu, dok se prilikom obrade podataka koristi i procesorski kapacitet senzorskih čvorova, koji često ostaje neiskorišćen.

Struktura teksta koji sledi

Tekst koji sledi prati generalnu strukturu naučnog rada koja je predložena u radu pod referencom [Milutinovic, 1996]. Prvo je dat pregled trenutnog stanja u oblasti, kroz analizu postojećih rešenja koja se mogu pronaći u otvorenoj literaturi, potom su dati opisi dva nova pristupa za mašinsko učenje u bežičnim senzorskim mrežama. Uz opise su predstavljeni i rezultati evaluacija koje su

rađene simulacionim analizama. Na kraju je dat osvrt na oblast kojom se ova teza bavi, uz autorova predviđanja daljih tokova istraživanja.

Naučni doprinos ove doktorske teze se sastoji iz više elemenata:

- Predložena je originalna klasifikacija rešenja za distribuirano mašinsko učenje u bežičnim senzorskim mrežama.
- Urađena je komparativna analiza različitih postojećih rešenja prema karakteristikama koje su prepoznate kao važne.
- Razvijen je novi algoritam (suštinskim unapređenjem već postojećeg) za vertikalno distribuiranu klasifikaciju u bežičnim senzorskim mrežama.
- Razvijen je novi algoritam za distribuirano grupisanje u bežičnim senzorskim mrežama, na osnovu primene Kohonenove neuralne mreže.

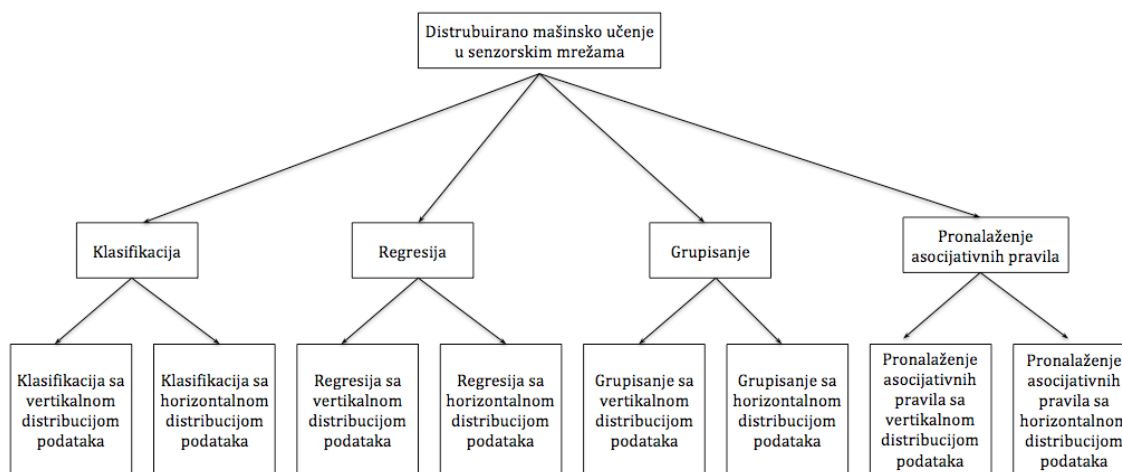
Nova predložena rešenja su detaljno analizirana putem simulacionih analiza, što je i predstavljeno u sklopu ove teze.

2. Pregled stanja u naučnoj oblasti

U ovome poglavlju će biti data šema za klasifikaciju postojećih rešenja vezanih za distribuirano mašinsko učenje sa primenom u bežičnim senzorskim mrežama. Data šema spada u originalne kontribucije ove teze, i uz nju će biti dato i obrazloženje odabranih kriterijuma po kojima su rešenja klasifikovana, diskusija opravdanosti ovih kriterijuma i njihove upotrebe na datom nivou klasifikacije.

Na najvišem nivou u klasifikacionom stablu (Slika 1) se nalazi podela koja je preuzeta iz tradicionalne klasifikacije algoritama mašinskog učenja. Prema ovoj podeli, algoritmi mogu biti algoritmi *grupisanja*, *pronalaženja asocijativnih pravila*, *klasifikacije* i *regresije*. Algoritmi koji pripadaju različitim kategorijama u ovoj klasifikacionoj šemi imaju različite primene, što ima bitan uticaj na njihov dizajn i principe rada. Prilikom odabira algoritma za neku aplikaciju, potrebno je prvo je definisati kojoj od ovih grupa aplikacija pripada, dok algoritmi iz drugih grupa nisu relevantni. Iz ovog razloga je data podela uzeta kao primarni, najviši stepen klasifikacije.

Niži nivo klasifikacije se bazira na izboru načina na koji se pretpostavlja da su podaci razdeljeni po senzorskim čvorovima: horizontalno ili vertikalno. Kod distribuiranih baza podataka princip distribucije je očigledan. Međutim, kod bežičnih senzorskih mreža ima prostora za različite interpretacije i pretpostavke (više detalja je dato u kasnijim sekcijama). Kako što će biti pokazano u tekstu koji sledi, problem distribucije podataka ima važne implikacije pri implementaciji i radu algoritama mašinskog učenja u bežičnim senzorskim mrežama, od kojih neki nisu očigledni na prvi pogled.



Slika 1. Klasifikaciono stablo pristupa distribuiranom mašinskom učenju sa primenom u bežičnim senzorskim mrežama. Na višem nivou klasifikacije pristupi su klasifikovani kao Grupisanje, Pronalaženje asocijativnih pravila, Klasifikacija i Regresija. Na nižem nivou klasifikacije pristupi su klasifikovani prema načinu distribucije podataka.

2.1. Kategorizacija tehnika mašinskog učenja

Kao što je već navedeno, algoritmi mašinskog učenja se mogu podeliti na algoritme grupisanja, pronalaženja asocijativnih pravila, klasifikacije i regresije. U ovom poglavlju data je kraća analiza karakteristika svake od ovih grupa.

Tehnike grupisanja (Clustering) pokušavaju da podele podatke u grupe (klastere) instanci koje su, po nekom kriterijumu, međusobno bliže u odnosu na instance koje se nalaze u drugim grupama. Kako je u pitanju tzv. nenadgledani tip učenja, grupisanje se ne oslanja na postojanje klasne labele, koja bi eksplicitno definisala moguće grupe i govorila kojim grupama podaci na osnovu kojih se uči pripadaju. U vezi sa bežičnim senzorskim mrežama, tehnike grupisanja se primenjuju kod naprednih strategija rutiranja [Akkaya, 2005], praćenja pokretnih meta [Chen, 2004], maksimizacije životnog veka mreže uz pokrivanje određene oblasti [Dasgupta, 2003], itd.

Cilj pronalaženja asocijativnih pravila je otkrivanje pravila kojima su podaci međusobno povezani. Ove tehnike mogu otkriti kompleksnije korelacije nego što je to slučaj sa običnim grupisanjem. U vezi sa bežičnim senzorskim

mrežama, tehnike pronalaženja asocijativnih pravila se mogu koristiti za poboljšanje pokrivenosti oblasti, analizom lokacija bitnih događaja i broja čvorova u mreži [Samarah, 2008], optimizaciju grupisanja i ponovno grupisanje na osnovu analize pravila samog grupisanja [Chong, 2008], kao i analizu performansi mreže i predviđanje budućih kvarova [Boukerche, 2007].

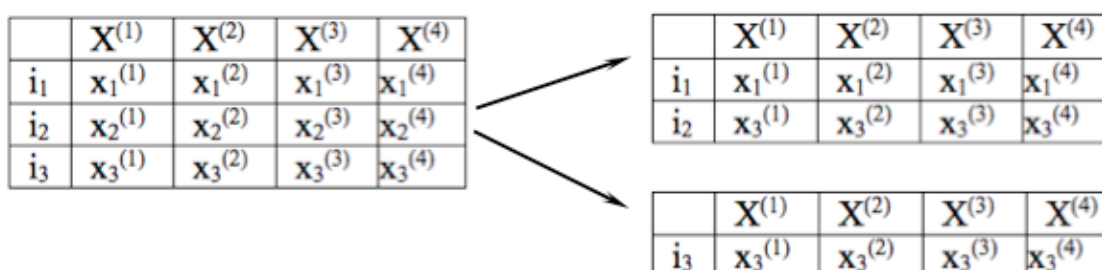
Klasifikacione tehnike služe da bi se podaci svrstali u neku od predefinisanih, diskretnih klasa. Klasifikacija spada u *nadgledane* metode, što znači da je potrebno da moguće klase podataka budu unapred definisane i poznate za podatke na osnovu kojih se vrši učenje. Drugim rečima, svaka instanca u skupu podataka koji se koristi za treniranje mora da ima pridruženu tzv. klasnu labelu, koja označava kojoj klasi podatak pripada. Klasifikatori se formiraju u fazi učenja, kada se klasne labele uzimaju u obzir, potom se, u fazi upotrebe, koriste da bi se odredila klasa drugim podacima, za koje klasna labela nije poznata. U vezi sa bežičnim senzorskim mrežama, tehnike klasifikacije se mogu koristiti za detekciju događaja [Baheropour, 2012], detekciju ekstremnih vrednosti podataka [Zhang, 2012] ili otkrivanje neautorizovanih upada u sistem [Moses, 2003].

Tehnike regresije služe za predviđanje ili određivanje vrednosti kontinualne promenljive na osnovu vrednosti drugih promenljivih koje su na raspolaganju. Kao što je to slučaj i sa klasifikacijom, u pitanju je nadgledana forma učenja. Drugim rečima, u skupu podataka koji se koristi za učenje je potrebno da postoji i informacija o vrednosti promenljive čija se vrednost određuje. U vezi sa bežičnim senzorskim mrežama, tehnike regresije se mogu koristiti za vremensku sinhronizaciju [Gao, 2007], poboljšanje preciznosti merenja [Otto, 2006], samolokalizaciju [Moses, 2003], itd.

2.2. Distribucija podataka

Veoma važna karakteristika distribuiranog algoritma mašinskog učenja se odnosi na način na koji su podaci distribuirani. Uzmimo skup podataka sa M različitih atributa i N različitih instanci. Ovakav skup se može posmatrati kao matrica dimenzija $N \times M$. Svaki red u matrici predstavlja jednu instancu podataka, i sastoji se od vektora vrednosti svih atributa. Svaka kolona predstavlja jedan atribut, i sadrži vrednosti tog atributa za sve instance u skupu podataka. Da bi algoritam mašinskog učenja bio distribuiran, potrebno je da su podaci sa kojima radi takođe distribuirani, odnosno podeljeni u podskupove, ili particije. Potom se delovi distribuiranog algoritma izvršavaju na pojedinačnim particijama.

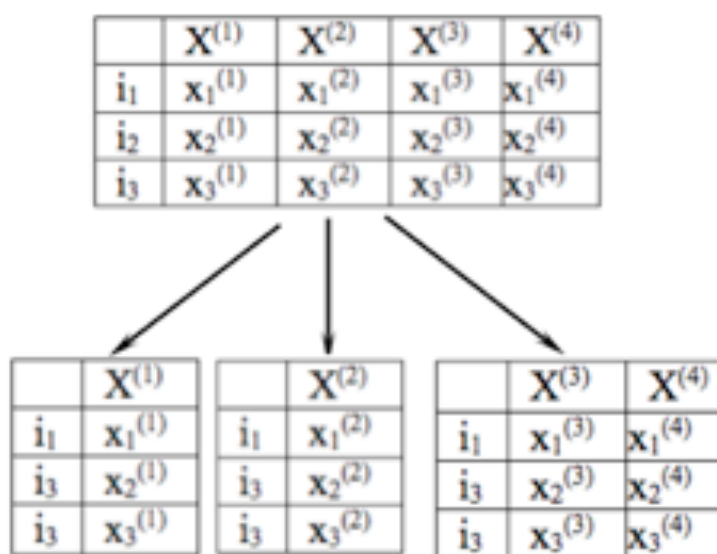
Podaci su distribuirani horizontalno ako su razdeljeni tako da svaka particija sadrži podskup skupa redova matrice. Drugim rečima, u svakoj particiji se nalaze vrednosti svih atributa, ali samo za podskup instanci (Slika 2). Horizontalna distribucija se ponekad naziva i *homogenom* distribucijom podataka.



Slika 2. Horizontalna distribucija podataka: svaka lokacija sadrži podskup instanci celokupnog skupa podataka, sa celim vektorima atributa. Legenda: $X^{(j)}$ –atribut j , i_k – instance k , $x_k^{(j)}$ – vrednost atributa j za podatak k

Podaci su vertikalno distribuirani ako su razdeljeni tako da svaka particija sadrži podskup skupa kolona matrice podataka. Drugim rečima, podaci su

vertikalno distribuirani ako se u svakoj particiji nalaze podaci o svim instancama podataka u skupu, ali se u pojedinačnim particijama nalaze različiti podskupovi atributa (Slika 2). Vertikalna distribucija se ponekad naziva i *heterogenom* distribucijom podataka.



Slika 3. Vertikalna distribucija podataka: svaka lokacija sadrži podskup atributa celokupnog skupa podataka, sa celim skupom instanci. Legenda: $X^{(j)}$ -atribut j , i_k - instance k , $x_k^{(j)}$ - vrednost atributa j za podatak k

U slučaju distribuiranog mašinskog učenja iz senzorskih mreža atribut u skupu podataka predstavlja očitavanje sa jednog senzora. Podaci u mreži su distribuirani po senzorskim čvorovima tako da se na jednom čvoru nalaze podaci koji su pročitani sa senzora koji se nalaze na tom čvoru. Postavlja se pitanje šta u ovakvom sistemu predstavlja jednu instancu u skupu podataka, i pri tome su moguća dva stanovišta.

Jedna instanca podataka se sastoji od skupa podataka koji su pročitani na jednom senzorskom čvoru u jednom trenutku. Ukoliko čvor npr. sadrži senzore za temperaturu, vlažnost i osvetljenost, jedna instanca predstavlja ove tri vrednosti uzete u nekom trenutku. Ovakav pogled je u skladu sa horizontalnom distribucijom podataka. Iako na prvi pogled sasvim logična, ova ideja unosi dve važne pretpostavke. Prva pretpostavka je da je mreža

homogena, odnosno da je svaki čvor opremljen istim brojem i tipovima senzora. Iako je u velikom broju slučajeva ova pretpostavka zadovoljena, to nije uvek slučaj. Druga pretpostavka je da vrednosti očitane na različitim čvorovima u mreži, širom cele mreže imaju ekvivalentno značenje i da su istoj (ili sličnoj) vezi sa promenama u okolini čvora.



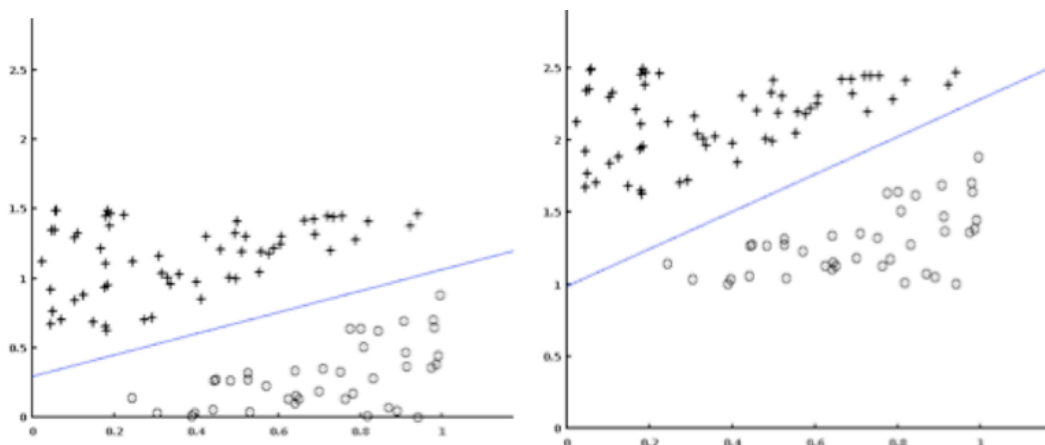
Slika 4: Bežična senzorska mreža za otkrivanje eksplozija. Senzori na čvoru X_1 otkrivaju velike promene usled eksplozije Ex_1 , ali ne detektuju nikakvu promenu pri eksploziji Ex_2 . Senzori na čvoru X_3 detektuju veliku promenu pri eksploziji Ex_3 , ali ne detektuju eksploziju Ex_1 . Senzori na čvoru X_2 detektuju male promene pri obe eksplozije.

Kako bismo ilustrovali drugu pretpostavku, uvešćemo sledeći, znatno uprošćeni primer: bežična senzorska mreža se koristi za otkrivanje eventualnih eksplozija u nekoj oblasti u kojoj je postavljena (primer je dat na Slici 4). U slučaju da se eksplozija označena kao Ex_1 dogodi, senzori smešteni na senzorskom čvoru X_1 bi detektovali veliki skok u temperaturi. Senzori na čvoru X_2 bi takođe detektovali temperaturni skok, ali u znatno manjoj meri, s obzirom na to da je ovaj čvor više udaljen od mesta eksplozije. Ukoliko bi se ova dva očitavanja posmatrala kao odvojene instance podataka, to bi rezultovalo nastankom dve instance: *veliko povećanje temperature => eksplozija* i *malo povećanje temperature => eksplozija*. Algoritam mašinskog učenja bi na osnovu toga mogao da izvuče hipotezu *bilo kakvo povećanje temperature => eksplozija*. Klasifikator baziran na ovakvoj hipotezi bi mogao da funkcioniše ispravno, ali

bi bio podložan šumu i dejstvu drugih faktora (malo povećanje temperature može imati i druge uzročnike iz okruženja).

Sa druge strane, ukoliko pretpostavimo vertikalnu distribuciju podataka, skup očitavanja sa svakog pojedinačnog senzora, sa svakog pojedinačnog čvora u mreži predstavlja zaseban atribut. U ovom slučaju jedna instanca podataka predstavlja vektor očitavanja sa svih senzora u mreži. Drugim rečima, vektor atributa se dobija tako što se spoje svi vektori koji sadrže očitavanja sa pojedinih čvorova u mreži. Na ranije predstavljenom primeru, uz vertikalnu distribuciju podataka, u mreži postoji samo jedna instanca za svaki vremenski trenutak. U trenutku eksplozije EX_1 ona bi izgledala: $D_1 = [\textit{veliko povećanje temperature kod } X_1, \textit{malo povećanje temperature kod } X_2, \textit{normalna temperatura kod } X_3]$. U trenutku eksplozije EX_2 ceo podatak bi bio: $D_2 = [\textit{normalna temperatura kod } X_1, \textit{malo povećanje temperature kod } X_2, \textit{veliko povećanje temperature kod } X_3]$. Iz ovakvih podataka neki klasifikator bi, na primer, mogao da zaključi da *veliko povećanje temperature kod bilo kog čvora* označava da se dogodila eksplozija, što je zaključak koji je mnogo bliži onome što želimo i očekujemo da saznamo iz mreža. Dodatno, ukoliko u sistem uvedemo mogućnost da razlikuje eksplozije prema lokaciji, sistem bi mogao i da određuje mesto gde se eksplozija dogodila.

Kako bismo dali nešto formalniju analizu, posmatraćemo još jedan primer. Mala senzorska mreža se sastoji od dva čvora. Neka svaki čvor sadrži po dva senzora, i neka su čvorovi međusobno identični (tipovi senzora koje sadrže su isti). Neka senzori prvog tipa daju potpuno ista očitavanja (na Slici 5 ova očitavanja su data na horizontalnoj osi). Međutim, neka se očitavanja koja daju senzori drugog tipa razlikuju između senzora na datim čvorovima, na primer tako što između njih postoji izvestan linearni ofset, tako da senzor na drugom čvoru uvek daje vrednost koja je za jednu jedinicu mere veća od očitavanja na senzoru na prvom čvoru (na Slici 5 očitavanja su data na vertikalnoj osi). Data situacija je u praksi moguća, čak veoma česta usled odstupanja u proizvodnom procesu ili nedovoljne kalibracije. Neka su sve ostale pretpostavke uvedene kao potrebne da bi se usvojila horizontalna distribucija podataka zadovoljene.



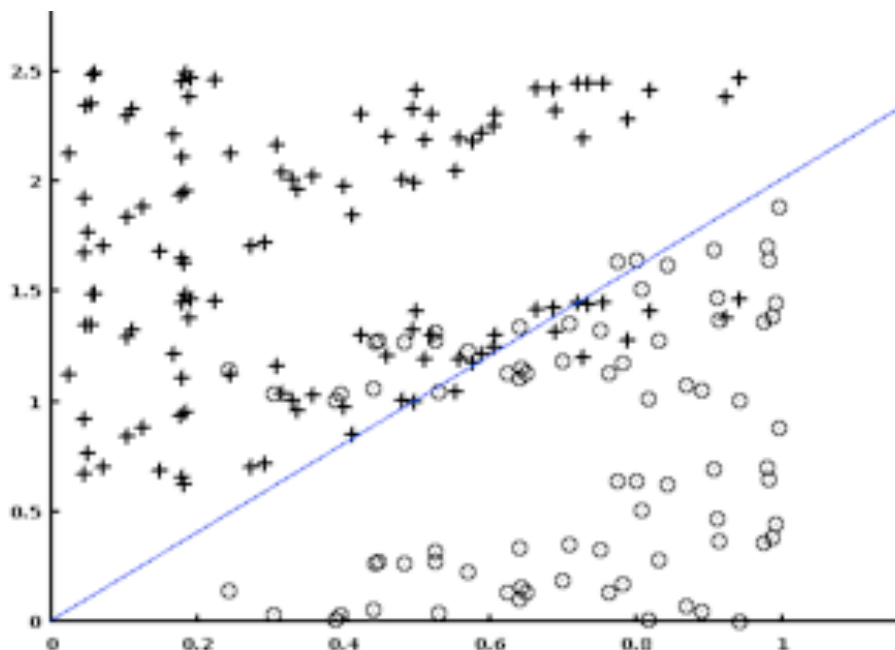
Slika 5. Razdvajajuće prave dobijene linearnom regresijom treniranom na dva hipotetička skupa podataka, dobijenih sa dva senzorska čvora, od kojih svaki sadrži po dva senzora. Očitavanja sa jednog para senzora istog tipa (po jednog senzora sa svakog čvora) su identična i predstavljena na horizontalnoj osi. Očitavanja sa drugog para senzora se razlikuju samo u linearnom ofsetu. Svako očitavanje je vezano ili za pozitivno stanje u okruženju (označeno kao +) ili za negativno stanje u okruženju (što je označeno kao o). 'Stanje u okruženju' predstavlja informaciju do koje se želi doći. U oba slučaja, linearnom regresijom je moguće naći liniju koja savršeno odvaja očitavanja, odnosno na osnovu vrednosti očitanih sa senzora predviđa u kojem stanju se okruženje nalazi.

Na Slici 5 su date linije dobijene linearnom regresijom kada su klasifikatori trenirani odvojeno na svakom od dva skupa podataka dobijenih sa dva senzorska čvora. Pošto je problem relativno jednostavan, veoma lako se dobija rešenje koje savršeno razdvaja podatke u zavisnosti od toga kojoj klasi pripadaju. Međutim, ukoliko se ova dva skupa podataka horizontalno spoje, rezultujući problem postaje linearno nerazdvojiv, što dovodi do loših performansi pri pokušaju upotrebe linearnog klasifikatora kao što je linearna regresija (Slika 6). Ukoliko se skupovi podataka spoje vertikalno, problem se premešta u tri dimenzije¹, što je prikazano na Slici 7. Kao što se može videti na Slici 7, u ovom slučaju je problem ponovo moguće rešiti linearnom regresijom, s

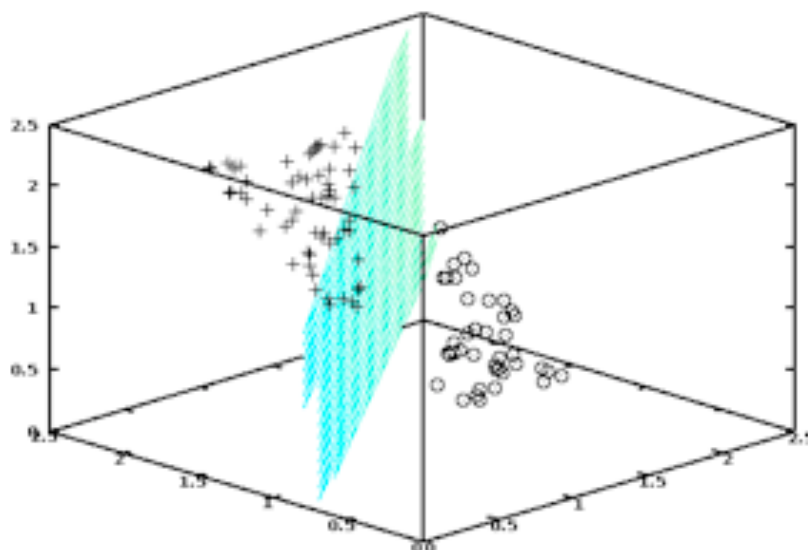
1

Napomena: ukoliko bi striktno pratili principe vertikalnog spajanja skupova podataka, rezultujući skup bi imao četiri dimenzije (po jednu za očitavanja sa svakog od senzora u mreži). Međutim, kako su očitavanja sa jednog para senzora u potpunosti identična, te dve dimenzije su u potpunosti redundantne, i mogu se svesti na jednu bez gubitka informacija.

tim da je rezultat prikazana razdvajajuća površ.



Slika 6. Razdvajajuća linija dobijena linearnom regresijom urađenom na istim podacima kao na Slici 5, ali pošto su skupovi podataka horizontalno spojeni. U ovom slučaju više nije moguće ostvariti savršeno linearno odvajanje klasa, pošto u podacima postoji varijacija koja nije posledica stanja u okruženju, već karakteristika senzora koji su posmatrani kao ekvivalentni.



Slika 7. Razdvajajuća ravan dobijena linearnom regresijom urađenom na istim podacima kao na Slici 5, ali pošto su skupovi podataka vertikalno spojeni. U ovom slučaju linearna regresija nalazi ravan kojom je moguće savršeno odvojiti klase u podacima.

Ipak, treba uzeti u obzir da i usvajanje vertikalne distribucije podataka sa sobom nosi određene mane. Prvi problem je u tome što vektor atributa postaje velik, i raste sa brojem senzora (ne samo senzorskih čvorova!) u mreži. Ovaj problem se donekle rešava algoritmima distribuiranog mašinskog učenja koji su sposobni da uče iz nekog podskupa svih atributa, ali ga svejedno treba uzeti u obzir.

Drugi problem je u tome što sada samo jedna instanca podataka zahteva očitavanja sa svih senzora u mreži, čime se znatno otežava prikupljanje skupa podataka koji je dovoljne veličine da bi se odradio metodama mašinskog učenja (nasuprot, kod horizontalnog spajanja skupova podataka svaki čvor u mreži generiše po jednu instancu podataka).

Treći problem nastaje zbog specifičnosti senzorskih mreža. Naime, čvorovi u senzorskim mrežama često „spavaju“ (nalaze se u režimu rada u kojem čuvaju bateriju tako što obustave funkcionisanje, ili samo komunikaciju), zbog čega u nekim vremenskim trenucima od nekih čvorova ne stižu podaci. Takođe, senzorski čvorovi se često kvare, ili ostaju bez energije, čime trajno ispadaju iz mreže. Kada se dogodi situacija da podaci sa nekog senzorskog čvora nisu dostupni, dolazi do nastanka tzv. nedostajućih vrednosti, koje predstavljaju ozbiljan problem u procesu mašinskog učenja (nasuprot, kod horizontalnog spajanja podataka došlo bi samo do manjeg broja instanci podataka u skupu). Dakle, ukoliko se očekuju „spavanje“ ili otkazi čvorova u mreži, mora se obezbediti da upotrebljeni algoritam za mašinsko učenje može da izađe na kraj sa nedostajućim vrednostima.

Konačno, četvrti problem kod vertikalnog spajanja skupova podataka leži u činjenici da ono zahteva neku vrstu vremenske sinhronizacije (barem na logičkom, ako ne i na fizičkom nivou). Sinhronizacija je neophodna kako bi se odgovarajuća očitavanja sa različitih čvorova mogla spojiti i vezati za isti vremenski trenutak (u ranije predstavljenom primeru to znači da je, na primer,

za svako čitanje sa sva tri čvora potreban i podatak da li je ono nastalo u trenutku eksplozije EX_1 , EX_2 ili u nekom sasvim drugom trenutku).

Hibridne distribucije podatka, koje kombinuju horizontalnu i vertikalnu distribuciju, takođe su moguće. Na primer, čvorovi se mogu grupisati na neki način, a potom se može uzeti da su podaci unutar jedne grupe distribuirani vertikalno, a između grupa horizontalno. Problemi vezani za dva osnovna načina distribucije podataka i dalje postoje, ali mogu biti delimično umanjeni, i izmešteni na nivo razmatranja unutar ili između grupa čvorova, umesto da se posmatraju individualni čvorovi u celoj mreži.

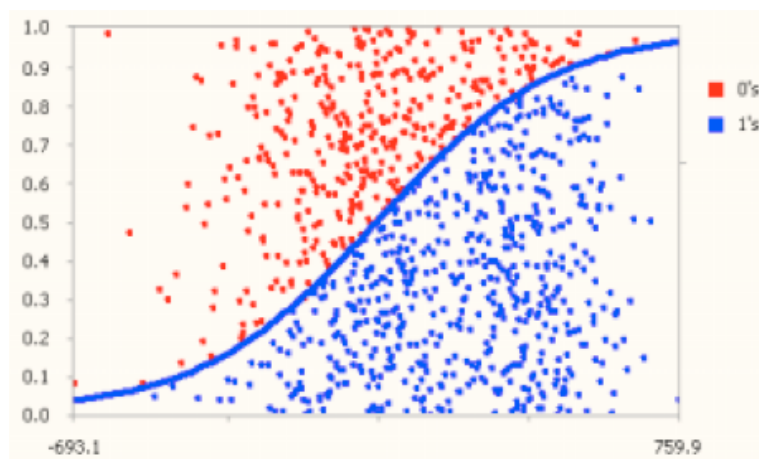
Ukratko, pogled na bežičnu senzorsku mrežu kao sistem sa horizontalnom distribucijom podataka predstavlja relativno jednostavan pristup koji zahteva da određeni skup uslova bude zadovoljen. Ti uslovi kod senzorskih mreža često nisu zadovoljeni. Sa druge strane, ideja vertikalne distribucije podataka daje generalniji pristup, koji donekle bolje odgovara prirodi problema kod bežičnih senzorskih mreža. Međutim, vertikalna distribucija unosi veći broj praktičnih problema koji nastaju i iz teorije informacija i iz specifičnosti bežičnih senzorskih mreža i koji moraju biti rešeni pri razvoju i implementaciji algoritma i sistema. Pri odabiru pristupa za novi algoritam sa određenom namenom, ukoliko su oba načina distribucije podataka moguća, prema Okamovoj oštrici, treba odabrati onaj jednostavniji, odnosno onaj koji predviđa horizontalnu distribuciju. U suprotnom, neophodno je razviti sistem sa vertikalnim spajanjem skupova podataka.

2.3. Postojeća rešenja za mašinsko učenje u senzorskim mrežama

U nastavku će biti dat pregled postojećih rešenja za distribuirano mašinsko učenje sa primenom u bežičnim senzorskim mrežama. Struktura teksta prati klasifikaciju datu na Slici 1: rešenja su grupisana prema datim kategorijama, i navedena u redosledu koji odgovara redosledu listova u klasifikacionom stablu, sa desne ka levoj strani.

2.3.1. Postojeća rešenja za klasifikaciju

Klasifikacija spada u takozvane *nadgledane* tehnike mašinskog učenja. Za razliku od nenadgledanog učenja, ulaz u nadgledani algoritam su podaci i posebna oznaka, takozvana labela, koja govori šta se traži, ili koju veličinu treba izračunati na osnovu datih podataka. Svakome podatku je pridružena po jedna vrednost labele. Kod klasifikacije, labela je kategoričkog tipa, odnosno označava jednu od nekoliko mogućih klasa kojoj dati podatak pripada. Cilj klasifikacije je pogoditi vrednost labele, odnosno odrediti kojoj klasi dati podatak pripada. Primeri mogu biti predviđanje da li je neka osoba zainteresovana za određenu temu ili proizvod, da li neki gen utiče na određenu bolest, itd. U smislu bežičnih senzorskih mreža, klasifikacija najčešće podrazumeva određivanje nekog stanja ili otkrivanje i definisanje nekog događaja u okruženju na osnovu vrednosti očitavanja sa senzora.



Slika 8: Klasifikacija logističkom regresijom

3.2.1.1 Klasifikacija sa vertikalno distribuiranim podacima

Jedan klasifikacioni algoritam za učenje iz vertikalno distribuiranih podataka je dat u radu označenom referencom [McConnell, 2008] (istraživanje je izvedeno na Računarskom fakultetu „Queens“ koledža). Pristup se bazira na proceduri većinskog glasanja na osnovu međurezultata koji su dobijeni zasebnim analizama podataka, pojedinačno na svakom od čvorova. Svaki čvor gradi lokalno stablo odlučivanja, i odluke koje su dobijene na osnovu ovih stabala se

kasnije prebrojavaju kako bi se došlo do globalne odluke. Kako algoritmu nije dato posebno ime, u ovoj tezi će biti označen po imenu prvog autora rada, odnosno kao Mekonel algoritam.

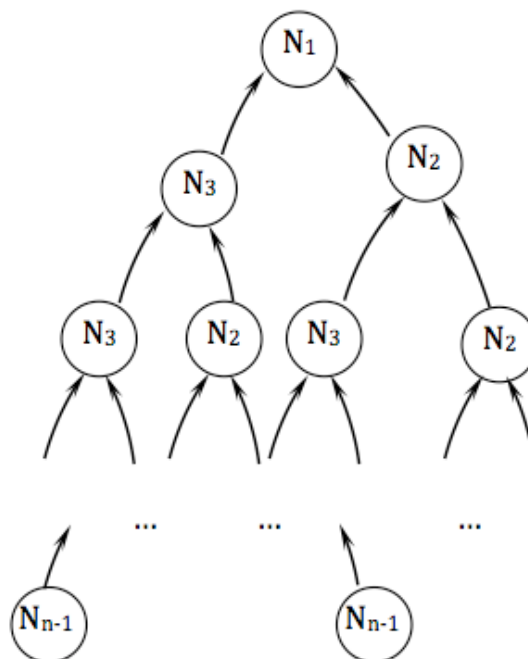
Pretpostavlja se da u sistemu postoji centralni server na kome postoji skup ranijih očitavanja iz cele mreže, zajedno sa podatkom o vrednosti promenljive čije je određivanje cilj analize (podatak u stanju ili događajima u okruženju). Podaci iz okruženja koji se žele pratiti (labele) čuvaju se u diskretizovanom obliku, odnosno uzimaju vrednosti iz nekog diskretnog skupa vrednosti koji opisuju okruženje na dovoljno precizan način.

U fazi treniranja labele se šalju na svaki čvor u mreži, uparene sa očitavanjima sa svih senzora koji se nalaze na datom čvoru. Ovako primljeni podaci sačinjavaju skup podataka na osnovu kojega se vrši treniranje lokalnih klasifikatora na svakom čvoru. Verzija algoritma koja je opisana i analizirana u radu koristi C4.5 stabla odlučivanja kao lokalne klasifikatore na čvorovima, ali isti princip se može primeniti i sa bilo kojim drugim lokalnim klasifikacionim algoritmom. Predstavljena su dva moguća pristupa treniranju sistema. U slučaju da kapaciteti čvorova nisu veoma ograničeni (pre svega u smislu raspoložive memorije) moguće je poslati ceo lokalni trening set odjednom. U suprotnom, podaci se mogu slati tako što se šalje i koristi po jedan podatak u svakom trenutku.

U fazi primene svaki čvor vrši očitavanja sa senzora koji se nalaze na njemu. Očitane vrednosti se propuštaju kroz lokalni klasifikator na samom čvoru i određuje se lokalna odluka o vrednosti labele. Ovako dobijene labele se iz cele mreže šalju do centralnog servera gde se većinskom logikom određuje vrednost globalne labele za celu mrežu. Dve varijante većinske logike su predložene. Prva varijanta je *jednostavno glasanje*, u kojem se prebrojavaju različite vrednosti labela stiglih sa različitih čvorova, i vrednost koja se najčešće javlja proglašava se za globalnu labelu. Druga varijanta predstavlja težinsko glasanje, pri čemu se

umesto prebrojavanja sabiraju težine koje su dodeljene u skladu sa poverenjem u tačnost rezultata koji stižu sa datih čvorova.

Mekonel algoritam je razvijen u pretpostavku da je potrebno da se bežična senzorska mreža posmatra kao sistem kod kojega su podaci vertikalno distribuirani. Međutim, isti pristup bi mogao da radi i u slučaju horizontalne distribucije podataka, kao i u hibridnom slučajevima. Slika 9 predstavlja dijagram toka podataka u mreži pri radu algoritma, dok Slika 10 daje pseudokod Mekonel algoritma.



Slika 9. Obrazac komunikacije u bežičnoj senzorskoj mreži u toku rada Mekonel algoritma.

<p>McConnell approach algorithm</p> <p>For each node do</p> <p> The node takes the readings from its local sensors.</p> <p> The node's local predictor is used with the readings to make a local prediction.</p> <p> The node sends the local prediction to the central server.</p> <p> If Voting scheme other than Simple voting is used</p> <p> Find the appropriate weight associated with the node and its local prediction.</p> <p> End If</p> <p>End For</p> <p>Sum the votes across all of the nodes to reach a global prediction.</p>
--

Slika 10. Pseudokod za Mekonel algoritam

Osnovna motivacija za razvoj Mekonel algoritma je bila smanjivanje neophodne komunikacije u radu bežične senzorske mreže. U tradicionalnom načinu funkcionisanje mreže svi podaci sa senzora na čvorovima u sistemu se šalju do centralne lokacije gde se obrađuju. Mekonel algoritam omogućava da se šalje samo onoliko bitova koliko je potrebno da bi se kodirale različite vrednosti labela koje dolaze iz lokalnih klasifikatora. Efikasnost algoritma time u velikoj meri zavisi od: 1) broja senzora na svakom čvoru, 2) broja bitova koji su neophodni da bi se kodirala očitavanja sa jednog senzora, 3) broja mogućih vrednosti koje labele iz lokalnih klasifikatora mogu da uzmu. Opisani algoritam je veoma pogodan za situacije kada je broj vrednosti lokalnih labela mali (okruženje može biti u nekoliko stanja, ili se očekuje neki od manjeg broja mogućih događaja), i kada čvorovi sadrže veći broj senzora visoke preciznosti.

Mekonel algoritam je u navedenom radu takođe proširen tako da može da funkcioniše i u situaciji kada vremenom dolazi do promena u zakonitostima koje povezuju očitavanja sa senzora i vrednosti labela lokalnih klasifikatora na čvorovima. Do ovakvih promena može doći ukoliko postoje fizičke promene u okruženju, ali i u slučaju kada je senzorski čvor pomeren na drugu lokaciju u odnosu na onu na kojoj je bio kada je nastao skup podataka korišćen za treniranje lokalnog klasifikatora. Pošto se lokalni klasifikatori upravo oslanjaju na pronalaženje zakonitosti koje povezuju očitavanja sa senzora i vrednosti labela i na osnovu njih predviđaju labele, jasno je da promene mogu imati katastrofalne posledice po rad sistema. Kako bi se problem rešio, sistem prati tačnost rada svakog pojedinačnog čvora, kao procenat slaganja sa konačnom globalnom odlukom. Ukoliko dođe do većih promena u ovome slaganju, ponovo se pokreće faza treniranja lokalnih klasifikatora, sa svežijim podacima.

Pored Mekonel algoritma, grupi vertikalno distribuiranih klasifikacionih rešenja za distribuirano mašinsko učenje sa primenom u bežičnim senzorskim mrežama pripada i algoritam opisan u referenci [Kokiopoulou, 2011] (istraživanje obavljeno na ETH, Cirih). Ovaj pristup, nazvan distMACS (*distributed*

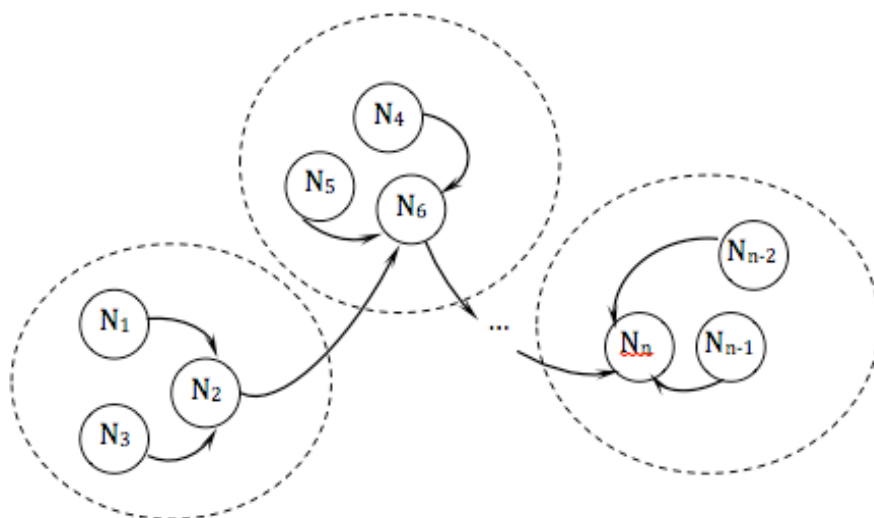
MANifold Smoothing under Constraints, (distribuirano višestruko izravnjavanje uz ograničenja), koristi kNN algoritam kao lokalne klasifikatore. Za razliku od Mekonel algoritma distMACS ne koristi centralnu lokaciju za donošenje globalne odluke, već se one donose na svakome od čvorova putem distribuiranog konsenzusa, kroz iterativnu proceduru.

2.3.1.2. Klasifikacija sa horizontalno distribuiranim podacima

U ovu klasu rešenja distribuiranih klasifikacionih rešenja za distribuirano mašinsko učenje sa primenom u bežičnim senzorskim mrežama spadaju prilagođene verzije *mašina sa vektorima podrške* (*Support Vector Machine*). Jedan primer je DFP-SVM algoritam opisan u referencama [Flouri, 2006] i [Flouri, 2008] (istraživanja su vršena na Univerzitetu na Kritu i Univerzitetu u Valensiji). Mašine sa vektorima podrške, koje se nekada nazivaju i *klasifikatori sa maksimalnim marginama*, predstavljaju klasu algoritama koji vrše klasifikaciju tako što postavljaju razdvajajuću hiperravan između instanci podataka koje pripadaju različitim klasama. Za razliku od linearne i logističke regresije koje uzimaju u obzir sve podatke, mašine sa vektorima podrške rade samo na osnovu podskupa instanci podataka različitih klasa koje su međusobno najbliže (ove tačke se nazivaju vektorima *podrške*). Distribuirana verzija algoritma za pravljenje klasifikatora sa vektorima podrške, predstavljena u referenci [Flouri, 2006], bazira se na opservaciji da se za dva različita skupa podataka dobija ista razdvajajuća hiperravan ukoliko su skupovi vektora podrške isti za oba skupa podataka. Drugim rečima, dovoljno je poznavati skup vektora podrške za neki skup podataka kako bi se pronašao optimalni klasifikator sa maksimalnom marginom. Iz ovoga proizilazi da je dovoljno da se u komunikaciji između čvorova u senzorskoj mreži prenose vektori podrške, dok ostali podaci nisu potrebni.

Na početku distribuiranog algoritma prilagođenog bežičnim senzorskim mrežama, opisanog u pomenutoj referenci [Flouri, 2006], čvorovi u mreži se razvrstavaju u grupe nekom od metoda topološkog grupisanja. Metode topološkog grupisanja formiraju grupe čvorova u skladu sa topologijom mreže, tako da su čvorovi koji su blizu jedni drugih u istim grupama. Osnovni motiv za ovakvo grupisanje leži u tome što je komunikacija između bliskih čvorova relativno jeftina. Sastavni deo grupisanja je i odabir jednog čvora koji se proglašava *vođom grupe*. Taj čvor preuzima zadatak da sinhronizuje celu grupu, komunicira sa ostalim grupama, itd. Za vođu grupe se često odabira čvor koji je fizički u sredini grupe, kako bi komunikacija sa njim za sve čvorove imala približno istu energetska cenu. Konkretna način podele mreže u grupe i odabira vođe grupe zavisi od metode koja je odabrana da bi se grupisanje izvršilo. Za princip rada algoritama klasifikacije putem vektora podrške koji trenutno opisujemo ovi detalji nisu od presudne važnosti, pa nećemo ulaziti dublje u moguće izbore za metodu topološkog grupisanja.

Pošto su svi čvorovi grupisani, svaki čvor šalje sve podatke koje očitava sa senzora do vođe grupe kojoj pripada. Na osnovu podataka sa svih čvorova u svojoj grupi vođa grupe kreira lokalni klasifikator sa vektorima podrške. Ovom procedurom vođa grupe određuje koji podaci, od svih podataka koje dobija, spadaju u lokalni skup vektora podrške. Zatim, vođe različitih grupa zamenjuju lokalne skupove vektora podrške, i ponovo treniraju lokalne klasifikatore, ovaj put uzimajući obzir samo svoje i dobijene vektore podrške, i na taj način određuju novi skup vektora podrške. Postupak razmene podataka između vođa grupa i pravljenja lokalnih klasifikatora (odnosno ponovnog određivanja skupova vektora podrške) se iterativno ponavlja do konvergencije skupa vektora podrške na nekom od čvorova. Slika 11 daje ilustraciju toka podataka.



Slika 11. Obrazac komunikacije u bežičnoj senzorskoj mreži u toku rada DFP-SVM algoritma.

DISTIRBUTED SVM TRAINING DFP-SVM	FIXED-PARTITION	WEIGHTED FIXED-PARTITION SVM TRAINING WDFP-SVM	DISTIRBUTED
Divide samples into clusters, where each cluster is of the same size. For every cluster do Define the cluster head (A sensor which receives data from all other sensors in the cluster, performs data fusion, and transmits the results to the base station.) Each sensor transmits its measurement sample vector to the cluster head. Cluster head combines the measurement sample vector with an estimation calculated (equations 1, 2) for the previous cluster head, to make a new SVM. Cluster head sends the estimated SVM to the cluster head that is the next one in order. End_For			
Notes: Cost function for estimation, equation for DFP-SVM: $\min_{\mathbf{w}, \xi} \Phi(\mathbf{w}, \xi) = \frac{1}{2} \ \mathbf{w}\ ^2 + C \sum_{i=1}^n \xi_i, \quad (1)$ where the parameter C defines the cost of constraint violation giving weight to measurements in the set.		Cost function for estimation, Equation for WDFP-SVM: $\Phi(\mathbf{w}, \xi) = \frac{1}{2} \ \mathbf{w}\ ^2 + C \left(\sum_{i \in I} \xi_i + L \sum_{i \in S} \xi_i \right), \quad (2)$ where the parameter L increases the cost for the old support vectors giving more weight to former measurements.	

Slika 12. Pseudokod za DFP-SVM algoritam

Unapređena verzija DFP-SVM algoritma, pod nazivom SSG-SVM koji je prilagođen za rad u bežičnim senzorskim mrežama u kojima dolazi do grešaka u komunikaciji je prikazan u radu pod referencom [Flouri, 2008].

Algoritmi DFP-SVM i SSG-SVM koriste prirodu klasifikatora sa vektorima podrške koja omogućava da se samo mali podskup podataka šalje i prima u toku rada. Međutim, ovakav pristup ispravno radi samo kada je bezbedno pretpostaviti da podaci prate horizontalnu distribuciju. Uz to, u nekim slučajevima nije moguće identifikovati skup vektora podrške kada su na raspolaganju samo lokalni podaci iz jedne grupe čvorova, usled čega dolazi do gubitaka nekih tačaka iz ovog skupa. Zbog ovog efekta, performanse DFP-SVM i SSG-SVM u mnogome zavise od distribucije podataka po grupama senzorskih čvorova. Da bi rešili ovaj problem, autori pomenutih algoritama su razvili dodatno unapređen algoritam [Flouri, 2009], koji uključuje i mogućnost da se u lokalni skup vektora podrške uključi širi skup tačaka, nego što lokalno deluje da potrebno (na primer, tačke koje su blizu razdvajajućoj hiperravni, ali ne spadaju u vektore podrške, odnosno postoje tačke koje su bliže od njih).

Prikazani algoritmi funkcionišu samo u slučaju da su potrebni linearni klasifikatori. U klasičnim primenama klasifikatori sa vektorima podrške se često kombinuju sa kernel funkcijama i tako se dobijaju veoma moćni algoritmi za rad sa nelinearnim podacima i zavisnostima. Međutim, trenutno ne postoje distribuirane kernel funkcije pogodne za primenu u bežičnim senzorskim mrežama, što svakako predstavlja jedan od pravaca budućih istraživanja.

2.3.2. Postojeća rešenja za regresiju

Regresija je vrsta mašinskog učenja pri kojoj se vrši analiza podataka u potrazi za zakonitostima pomoću kojih je moguće predvideti (približnu) vrednost neke kontinualne promenljive. Osnovna razlika u odnosu na klasifikaciju je kontinualnost promenljive koja se predviđa (labele). Pošto se bazira na određivanju vrednosti poznate promenljive, kao i klasifikacija, regresija spada u takozvane *nadgledane* tehnike mašinskog učenja. U ranijim poglavljima su pominjana dva konkretna algoritma koji u svome nazivu sadrže reč regresija: linearna regresija i logistička regresija. Treba napomenuti da su u kontekstu u

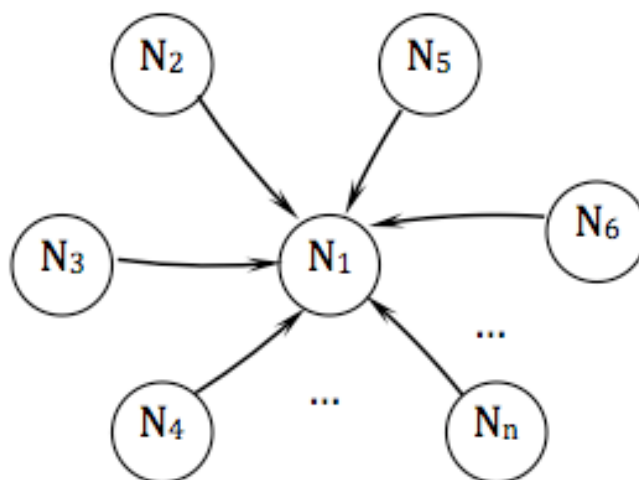
kojem su pominjani ovi algoritmi korišćeni u svrhu klasifikacije, a ne regresije u užem smislu. Štaviše, algoritam logističke regresije je isključivo klasifikacioni algoritam, dok se linearna regresija može koristiti i za regresiju i za klasifikaciju. U vezi sa bežičnim senzorskim mrežama regresija se može koristiti za predviđanje vrednosti očitavanja sa nekih senzora, ukoliko su poznata očitavanja drugih senzora u mreži.

2.3.2.1. Regresija sa vertikalno distribuiranim podacima

Primer algoritma regresija nad vertikalno distribuiranim podacima u bežičnim senzorskim mrežama je dat u radu pod referencom [Papadimitriou, 2009] (istraživanje vršeno na Univerzitetu Karnegi Melon). Algoritam pod nazivom SPIRIT (*Streaming Pattern Discovery in Multiple Timeseries*, Pronalaženje tekućih šablona u višestrukim vremenskim serijama) inkrementalno pronalazi korelacije i *skriveno promenljive* u numeričkim tokovima podataka. Pojam skrivena promenljiva u mašinskom učenju označava logičke promenljive čija se vrednost ne može izmeriti, već samo odrediti na osnovu vrednosti drugih promenljivih. Skriveno promenljive su često fiktivne, i mogu se uporediti sa međurezultatima kod klasičnog računanja. U slučaju SPIRIT algoritma, skrivene promenljive se postavljaju između ulaznih podataka sa senzora i labela čija se vrednost želi odrediti, tako da se iz sirovih podataka prvo određuju vrednosti skrivenih promenljivih, a zatim iz njih određuju labele.

Formalno predstavljen princip rada SPIRIT algoritma je sledeći: neka je dat skup od n polukonačnih tokova podataka, koji zajedno evoluiraju (trendovi u vrednostima podataka su korelisani). Neka ovi tokovi generišu vrednosti X_{tj} , gde je t vremenski trenutak, $t = 1, 2, \dots$, a j oznaka toka koji je generisao podatak ($1 \leq j \leq n$). Algoritam prvo određuje optimalan broj skrivenih promenljivih (neka je ovaj broj označen sa k), pomoću kojih se mogu opisati glavni trendovi u datim tokovima podataka. Potom se određuju težinski faktori w_{ij} sa kojim j -ti tok podataka utiče na vrednost i -te skriveno promenljive ($1 \leq j \leq n$, $1 \leq i \leq k$), tako da skrivene promenljive daju tačan sažetak celog skupa tokova podataka.

Posle početnog određivanja vrednosti, algoritam prati vrednosti skrivenih promenljivih i podataka koji dolaze iz tokova podataka i po potrebi podešava vrednosti težinskih faktora. Zahvaljujući ovim podešavanjima u toku rada, SPIRIT algoritam može da radi i u uslovima dinamičkih mreža kod kojih se topologija menja u vremenu ili dolazi do fizičkog pomeranja čvorova.



Slika 13. Obrazac komunikacije u bežičnoj senzorskoj mreži u toku rada SPIRIT algoritma.

SPIRIT algorithm	
Initialize the k hidden variables W to unit vectors $w_1 = [10 \dots 0]^T$, $w_2 = [010 \dots 0]^T$, etc. Initialize d_i ($i = 1, \dots, k$) to a small positive value.	
While x_{t+1} arrives do	
Update $x'_1 = x_{t+1}$.	
For $1 \leq i \leq k$ do	
Calculate $y_i, d_i, e_i, w_i, x_{t+1}$.	
End_For	
Update $x'_{i+1} = x_i - y_i w_i$.	
End_While	
$y_i = w_i^T x'_i$	($y_{t+1,i} =$ projection onto w_i)
$d_i = \lambda d_i + y_i^2$	(energy \propto i -th eigenval. of $X^T X$)
$e_i = x'_i - w_i y_i$	(error, $e_i \perp w_i$)
$w_i = w_i + w_i e_i / d_i$	(update PC estimate)

Slika 14. Pseudokod za SPIRIT algoritam

Autori predlažu nekoliko namena za SPIRIT algoritam. Jedna namena je predviđanje vrednosti ciljanih podataka (postavljenih labela, odnosno informacija koje se žele saznati na osnovu očitavanja senzora). Ovo se postiže na

osnovu toga što SPIRIT algoritam uzima u obzir podatke kroz dve dimenzije: vremensku, gde koristi korelacije između podataka u više vremenskih trenutaka, i „prostornu“, gde koristi korelacije između podataka pristiglih sa više različitih senzora. Kako postoje podaci o korelacijama u vremenu, moguće je predvideti vrednosti skrivenih promenljivih u nekom budućem trenutku.

Druga moguća namena je rešavanje problema vrednosti koje nedostaju, odnosno situacija u kojima sa nekog od senzorskih čvorova ne pristignu podaci. Pošto postoje podaci o korelacijama između očitavanja sa različitih senzora, moguće je približno odrediti vrednosti skrivenih promenljivih, čak i ukoliko neka od očitavanja nedostaju.

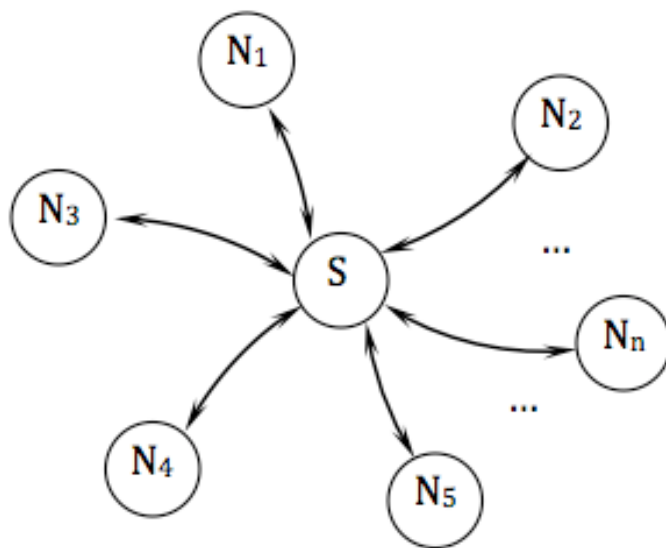
Konačno, autori predlažu da se SPIRIT algoritam može koristiti za interpretaciju podataka. To bi se moglo postići takozvanom statističkom analizom medijacije. Drugim rečima, prvo se utvrdi važnost dejstva neke skrivene promenljive na labelu, a potom i važnost dejstva pojedinih senzora na datu skrivenu promenljivu (ove informacije su praktično date vrednostima težinskih faktora). Na osnovu ovih podataka je moguće odrediti koja senzorska očitavanja i koliko utiču na labele. Rad SPIRIT algoritma je analiziran na podacima iz nekoliko skupova podataka, kao što su temperaturna merenja, merenja intenziteta svetla i koncentracije hlora u vodovodu, i uspešno je primenjen na pronalaženje glavnih trendova u podacima.

2.3.2.2. Regresija sa horizontalno distribuiranim podacima

U radu pod referencom [Goel, 2011] opisan je jedan primer regresionog algoritma za rad sa horizontalno distribuiranim podacima, pod nazivom PREMON (istraživanje vršeno na državnom univerzitetu i Nju Džersiju). Cilj ovog istraživanja je bio umanjivanje potrebne komunikacije u bežičnoj senzorskoj mreži. Centralna ideja se bazira na treniranju regresionog modela na centralizovanoj lokaciji za celu mrežu, kojim se na osnovu nekih od poznatih vrednosti očitanih sa senzora predviđa vrednost preostalih. Predikcije iz ovog

modela se potom mogu koristiti, umesto da se šalju i primaju sva očitavanja u mreži. Proširena verzija algoritma uvodi veći broj lokacija na kojima se treniraju regresioni modeli na osnovu podskupova senzorskih očitavanja, umesto samo jednog mesta za celu mrežu. U ovoj verziji algoritma na samom početku se vrši grupisanje čvorova, slično DFP-SVM algoritmu, a regresioni modeli se treniraju na čvorovima koji su odabrani kao vođe grupa. Verzija algoritma koja je proširena na ovaj način poboljšava skalabilnost rešenja, ali na drugi način ne utiče na princip rada.

Prediktivni regresioni model koji se koristi kod PREMON algoritma je jednostavna metoda linearne aproksimacije na osnovu prethodna dva vremenska očitavanja sa istog senzora. Pošto su kreirani, regresioni modeli se prebacuju i na svaki od čvorova, gde se uporedo sa očitavanjima sa senzora vrši i predikcija vrednosti koje se čitaju. Ukoliko se stvarna (očitanja) vrednost sa senzora razlikuje od one koju je predvideo regresioni model, čvor se uključuje u komunikaciju i šalje pravu očitavanu vrednost. Ukoliko razlika nije značajna, čvor ne šalje očitavanje sa senzora, pošto je isti regresioni model na raspolaganju i na centralnoj lokaciji, pa je procena vrednosti koja je tamo dobijena iz modela takođe bliska tom stvarnom očitavanju. Na Slici 15 je prikazan tok podataka u mreži u toku rada PREMON algoritma, dok Slika 16 daje implementaciju algoritma u pseudokodu.



Slika 15. Obrazac komunikacije u bežičnoj senzorskoj mreži u toku rada PREMON algoritma.

Rezultati eksperimenata koji su opisani u radu pokazuju da PREMON pristup znatno umanjuje potrošnju energije u bežičnim senzorskim mrežama, čime se produžava autonomija rada senzorskih čvorova, odnosno vreme rada cele mreže. Efikasnost rešenja u velikoj meri zavisi od predvidljivosti vrednosti koje se očitavaju sa senzora. Mogući pravac daljeg istraživanja u ovom domenu predstavlja upotreba kompleksnijih algoritama baziranih na analizi vremenskih serija, ako i primena nelinearnih modela.

The PREMON algorithm
At the base-station For each node do Take 4 most recent consecutive readings (X_1 , X_2 , X_3 , and X_4) Check X_1 and X_2 for changes in readings If change is detected Make a linear model by extrapolation of the change trend to readings X_3 and X_4 Calculate the difference between the model estimate and the actual values If error is below a threshold Assume model is valid Send the model to the corresponding node Else Discard the model End_If End_If End_For
At the node: Take sensor readings If a valid prediction model exists

```

Calculate the difference between the model estimate and the actual values
If error is below a threshold
    Take no action
Else
    Mark the model as invalid
    Send readings to the base station
End_If
Else
    Send readings to the base station
End_If

```

Slika 16. Pseudokod za PREMON algoritam.

U ovu grupu rešenja spada i algoritam pod nazivom TREG [Banerjee, 2005] (istraživanje vršeno u OBR Centru za distribuirano i mobilno računarstvo, Univerzitet u Sinsinatiju), koji prati relativno sličnu osnovnu ideju kao i opisani PREMON. U TREG algoritmu na početku rada se unutar bežične senzorske mreže formira podmreža koja ima strukturu binarnog stabla i koja pokriva oblast cele senzorske mreže. Čvorovi koji nisu u ovome stablu šalju svoje podatke i koordinate do najbližeg čvora koji jeste u stablu i koji predstavlja list. Listovi stabla potom prave regresione modele (regresioni model koji je analiziran u radu je baziran na polinomijalnoj regresiji drugog reda). Ulazi u regresiju su očitavanja sa različitih senzorskih čvorova i koordinate odgovarajućih čvorova. Proračunati regresioni koeficijenti se šalju uz stablo, prema korenom čvoru.

Čvorovi stabla koji nisu listovi čekaju da prime regresione koeficijente od dva čvora koja se nalaze neposredno ispod njega. Na osnovu ovih koeficijenata i podataka koji su na raspolaganju na datom čvoru kreira se novi, jedinstveni regresioni model koji se šalje dalje uz stablo. Procedura se ponavlja dok se u korenom čvoru stabla ne napravi konačni model. Ovim modelom se mogu aproksimativno određivati očitavanja iz cele mreže.

Alternativni pristup regresiji, takozvana *kernel regresija* [Scholkopf, 2002] je još jedna grupa algoritama mašinskog učenja koji su prilagođeni za distribuiranu

upotrebu u bežičnim senzorskim mrežama [Guestrin, 2004], [Predd, 2006], [Honeine, 2008].

Pristup na osnovu distribuirane kernel regresije za sisteme sa ograničenim kapacitetom za komunikaciju je predstavljen u radu pod referencom [Predd, 2006] (istraživanje vršeno na Univerzitetu Princeton). Metoda koristi činjenicu da je moguće konstruisati globalnu kernel regresiju, ukoliko postoji preklapanje u lokalnim skupovima podataka koji se nalaze na senzorskim čvorovima. Za podatke koji su isti na dva ili više čvorova, ti čvorovi iterativno iznova izračunavaju lokalnu kernel funkciju (pri čemu se koristi ceo lokalni skup podataka, na svakom čvoru odvojeno), i pri tome modifikuju vrednosti veštačkih labela koje su pridružene deljenim podacima. Iterativna procedura se izvršava na jednom čvoru, zatim se modifikovane vrednosti labela prosleđuju do sledećeg čvora koji deli isti podatke, gde se procedura ponavlja. Iteracija se završava kada svi čvorovi koji sadrže neki podatak po jednom izračunaju i izmene vrednosti veštačkih labela. Ovim se postiže da svaki čvor ima sopstveni regresioni model koji je prilagođen lokalnim podacima, ali se izvestan broj deljenih podataka koristi kako bi se obezbedila globalna koherencija (detaljnija analiza teoretskih uslova pod kojima je princip rada algoritma validan, kao i dokaz konvergencije dati su u referisanom radu). U nastavku teksta ovaj algoritam će biti označen kao KREG.

Slično rešenje, koje omogućava da se koristi kernel nižeg reda, čime se smanjuju kompleksnost pri izračunavanju i potrebe za komunikacijom, opisano je u radu pod referencom [Honeine, 2008]. U ovom pristupu u mreži se formira samo jedan kernel, ali na distribuiran način. Jedan od čvorova formira početni kernel, samo na osnovu lokalnih podataka. Kernel se potom prosleđuje do sledećeg čvora u mreži. Odabir početnog čvora i redosled čvorova nisu bitni, pa ih je moguće odabrati nekom metodom kojom se optimizuje putanja kroz celu mrežu, čime bi se, kao što će postati očigledno, optimizovala i količina komunikacije, a time i potrošnja energije. Pošto čvor primi kernel, vrši procenu

koliko precizno trenutni kernel predviđa podatke koji se nalaze u njegovom lokalnom skupu podataka. Ukoliko je maksimalna greška veća od predefinisane granice, čvor proširuje kernel. U suprotnom, kernel se dalje prosleđuje u neizmenjenom obliku. U nastavku teksta ovaj algoritam će biti označen kao RO-KREG.

2.3.3. Postojeća rešenja za grupisanje

Grupisanje (koje se često naziva i *klasterovanje*, tj. *clustering*) spada u takozvane *nenadgledane* tehnike mašinskog učenja. To znači da su ulaz u algoritam podaci bez posebne oznake koja govori šta se traži, ili koju veličinu treba izračunati na osnovu datih podataka. Cilj grupisanja je da ovakve podatke podeli u grupe koji su unutar grupe sličniji jedni drugima nego što je to slučaj kada se porede sa podacima iz neke druge grupe. Primeri mogu biti grupe ljudi koji imaju slične skupove interesovanja ili grupe gena koji imaju sličnu funkciju u organizmu.

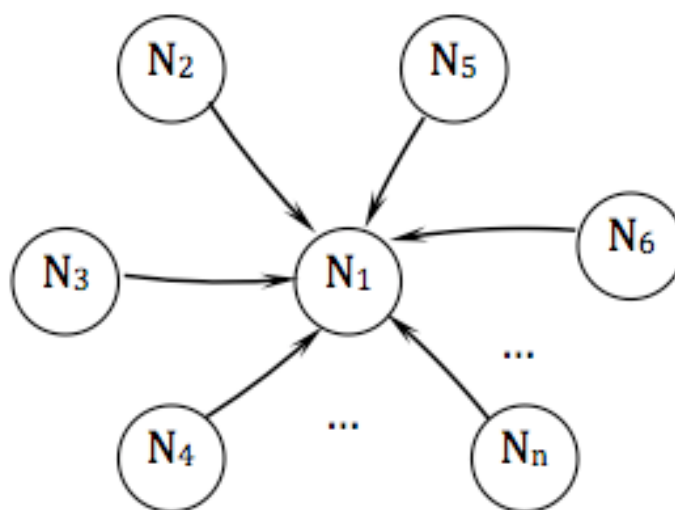
U vezi sa bežičnim senzorskim mrežama dosta je rađeno na grupisanju senzorskih čvorova. Cilj takvih rešenja je da se izvrši podela senzorske mreže u manje celine, pri čemu se najčešće u obzir uzimaju topologija mreže, obrasci komunikacije u mreži, ili pokrivenost oblasti u kojoj mreža radi. Primeri upotrebe grupisanja senzorskih čvorova su navedeni kod opisa algoritama klasifikacije i regresije, DFP-SVM i PREMON. Za razliku od ovakvih pristupa, gde se grupišu senzorski čvorovi, u nastavku će biti razmatrani algoritmi koji se bave grupisanjem *podataka* koje se očitavaju sa senzora u senzorskim mrežama.

2.3.3.1. Grupisanje sa vertikalno distribuiranim podacima

Jedan od primera algoritama grupisanja nad vertikalno distribuiranim podacima je dat u radu pod referencom [Johnson, 1999]. Ovaj algoritam nije razvijan specifično za rad u bežičnim senzorskim mrežama, ali niska kompleksnost izračunavanja i mali zahtevi za komunikacijom ga čine

pogodnim za takvu primenu, pri čemu ostale karakteristike ne postavljaju zahteve koje bi bilo teško ispuniti u senzorskoj mreži. Prema tome, ovaj algoritam bi se veoma lako mogao implementirati u bežičnoj senzorskoj mreži. Algoritam nosi naziv DHC (*Distributed Hierarchical Clustering*, odnosno *Distribuirano Hijerarhijsko Grupisanje*).

U prvoj fazi rada DHC algoritma svaki čvor grupiše lokalne podatka prema hijerarhijskom grupisanju. Potom se sve lokalne šeme grupisanja prenose na jednu centralnu lokaciju, gde se lokalni modeli spajaju u jedan globalni model. Globalni model se gradi tako što se izračunavaju mere međusobnog odstojanja instanci podataka prema svim lokalnim šemama grupisanja. Potom se ova odstojanja koriste za generisanje globalne klasifikacione šeme. Na Slici 17 je prikazan tok podataka u mreži u toku rada DHC algoritma, dok Slika 18 daje implementaciju algoritma u pseudokodu.



Slika 17. Obrazac komunikacije u bežičnoj senzorskoj mreži u toku rada DHC algoritma.

The DHC algorithm
At the base-station Receive all of the local models from the nodes For each selected data point do For each local model do Compute the upper and lower distance bounds between the data point and all of the other data points

```

End_For
Compute the average distance metric between the data point
and all of the other data points
Track the minimum distance as:
N(i) = index of the closest data point
D(i) = distance to the closest data point
End_For
Repeat n-1 times
Cluster1 = indexOf(min(D))
Cluster2 = N(Cluster1)
Agglomerate(Cluster1, Cluster2)
End_Repeat

```

Slika 18. Pseudokod za DHC algoritam.

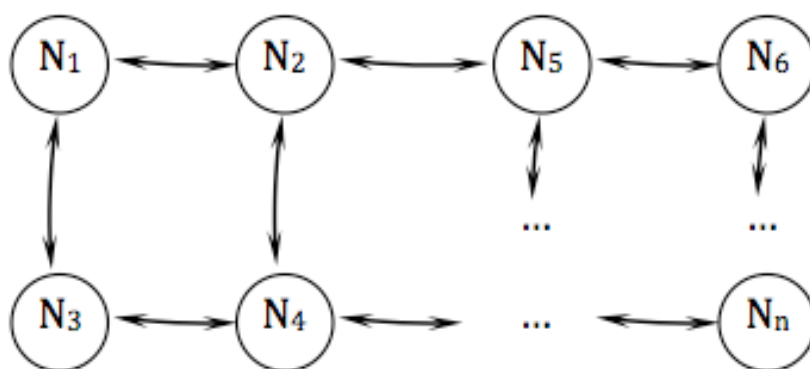
2.3.3.2. Grupisanje sa horizontalno distribuiranim podacima

Među algoritmima grupisanja, jedan od najjednostavnijih, ali često veoma efikasan je takozvani *K-means* algoritam. *K-means* se bazira na iterativnoj proceduri, u kojoj se grupe formiraju oko skupa virtuelnih tačaka u centru svake grupe, takozvanih centroida. Svaka iteracija algoritma sadrži dve faze: izračunavanje koordinata centroida (pri čemu se centroidi pomeraju u tačku koja je aritmetička sredina podataka koji su trenutno smešteni u pripadajuću grupu) i ponovno određivanje grupe kojoj podaci pripadaju (pri čemu se svaki podatak dodeljuje grupi oko trenutno najbližeg centroida). Na početku se centroidi postavljaju slučajnim izborom.

Modifikacija *K-means* algoritma koja omogućava rad u distribuiranom okruženju, kao što su bežične senzorske mreže, opisana je u radu pod referencom [Karagupta, 2006] (istraživanje je izvršeno na Univerzitetu u Merilendu). Algoritam počinje sa radom na jednom čvoru koji definiše početni skup centroida, i šalje ih svojim direktnim susedima. Ti čvorovi potom prosleđuju centroide svojim direktnim susedima, i tako dalje sve dok se početne vrednosti centroida ne propagiraju kroz celu mrežu, kada prva iteracija izračunavanja može da počne.

U toku izračunavanja čvorovi prikupljaju trenutne vrednosti centroida od svojih direktnih suseda. Po prijemu, čvorovi određuju nove vrednosti lokalnih

centroida, korišćenjem starih vrednosti lokalnih centroida, vrednosti koje su stigle od okolnih čvorova, i lokalnog skupa podataka (očitavanja sa senzora) koji se nalaze na čvoru. Izračunavanje u ovoj fazi se izvršava u dva koraka, slično kao osnovna verzija K-means algoritma, s tim što se kao podaci koriste lokalni skup podataka i vrednosti centroida dobijene sa okolnih čvorova. Svaki od čvorova nezavisno obustavlja iterativnu proceduru kada promena vrednosti lokalnih centroida padne ispod neke predefinisane vrednosti. Na Slici 19 je prikazan tok podataka u mreži u toku rada distribuiranog K-means algoritma, dok Slika 20 daje implementaciju algoritma u pseudokodu.



Slika 19. Obrazac komunikacije u bežičnoj senzorskoj mreži u toku rada distribuiranog K-means algoritma.

The distributed K-means algorithm
While termination criteria not reached do Pool all neighbors for their current cluster centroids Upon reception $dataset = local_data \cup received_centroids$ For_each X in $dataset$ do Find closest local centroid Assign X to the appropriate cluster End_For For each local centroid do Recalculate the centroid as the mean of all the data points assigned to the corresponding cluster End_For End_While

Slika 20. Pseudokod za distribuirani K-means algoritam.

Konvergencija i granice tačnosti (u smislu poklapanja rezultata sa osnovnom, centralizovanom verzijom) nisu matematički izvedene, dok empirijski testovi pokazuju da algoritam relativno brzo konvergira, sa visokim stepenom tačnosti.

2.3.4. Postojeća rešenja za pronalaženje asocijativnih pravila

Pronalaženje je skup nenadgledanih metoda mašinskog učenja koje su fokusirane na pronalaženja pravila koje korelišu instance podataka u skupu. U vezi sa bežičnim senzorskim mrežama ove metode se uglavnom primenjuju u cilju pronalaženja pravila kojima je moguće predvideti vrednosti očitane sa senzora na jednom čvoru, kada su poznate vrednosti očitane na ostalim čvorovima.

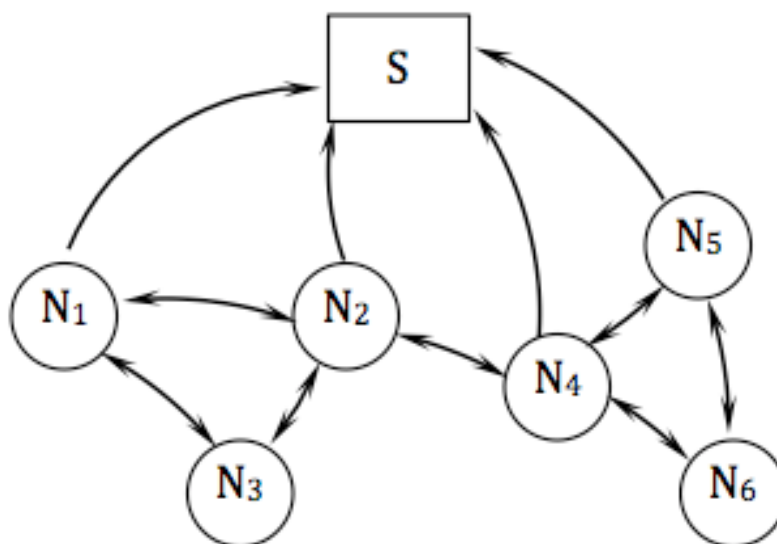
2.3.4.1. Pronalaženje asocijativnih pravila sa vertikalno distribuiranim podacima

Radovi pod referencama [Halatchev, 2005] i [Gruenwald, 2007] sadrže rešenje iz skupa metoda pronalaženja asocijativnih pravila sa vertikalno distribuiranim podacima (istraživanja su vršena na Univerzitetu u Oklahomi). Algoritam, pod nazivom FARM je razvijen sa ciljem približnog određivanja vrednosti koje nedostaju, kasne ili sadrže grešku.

FARM algoritam se bazira na modifikaciji poznatog Apriori algoritma. Kako bi se princip rada prilagodio protočnoj prirodi podataka u senzorskim mrežama, kao i umerenim procesorskim kapacitetima senzorskih čvorova, u obzir se uzimaju samo interakcije drugog reda među podacima (drugim rečima, analiziraju se samo zavisnosti koje se mogu pronaći kada se posmatraju parovi podataka, dok se zanemaruju zavisnosti koje se mogu pronaći ako se zajedno analiziraju veći brojevi instanci podataka). Druga modifikacija je uvođenje pokretnog prozora, čime se u obzir uzima samo predefinisani broj poslednjih očitavanja u mreži.

Ukratko, FARM algoritam radi na sledeći način: skorašnja očitavanja sa čvorova se smeštaju u bafer. Potom se ove vrednosti postavljaju u kocku, pri čemu se povezuju istovremena pojavljivanja različitih vrednosti na različitim čvorovima. Ukoliko u kasnijem radu vrednosti sa nekog od čvorova ne stignu, uzima se

vrednost za koju se utvrdi da je najverovatnije da se javi u kombinaciji sa vrednostima koje su stigle sa ostalih čvorova. Na Slici 21 je prikazan tok podataka u mreži u toku rada distribuiranog FARM algoritma, dok Slika 22 daje implementaciju algoritma u pseudokodu.



Slika 21. Obrazac komunikacije u bežičnoj senzorskoj mreži u toku rada distribuiranog FARM algoritma.

Iako opisani algoritam nije u potpunosti distribuiran, s obzirom na to da se celokupna obrada podataka izvršava na centralizovanoj lokaciji, princip rada se lako može prilagoditi distribuiranom izvršavanju, tako što se redukovane verzije pokrenu na samim čvorovima u mreži (redukovane u smislu da rade na podacima koji dolaze sa podskupova obližnjih čvorova), i ovime vrednosti koje nedostaju ili sadrže greške dodaju ili ispravljaju unutar senzorske mreže. FARM algoritam proširen na ovakav način bi zahtevao nešto veći stepen komunikacije u mreže u situacijama u kojima neki podatak ne postoji, pošto bi se vrednost tog podatka odredila u mreži i dalje transportovala kroz nju. Sa druge strane, opterećenje na centralnoj lokaciji bi se smanjilo, što povlači bolju skalabilnost rešenja.

FARM Algorithm
For (all rounds of sensor readings) do

```

begin
checkBuffer();
update();
estimateValue();
end.
checkBuffer()
While (the current session lasts) do
record the data received from a particular sensor to corresponding field in the Buffer
For (all fields in the Buffer) do
check if there is a missing value
If missing value exists
estimateValue() for that missing value
Else
send OK signal to queries
update()
End_If
* The Buffer is the data structure to store the arriving readings associated with the
corresponding sensors.

update()
// The purpose of this algorithm is to update the Cube and the Counter every time a new
round
(without missing values) of sensor readings is stored in the Buffer.
For all sensor readings in the Buffer do
update 1-itemsets
add new nodes at the front of the Cube
discard the oldest nodes at the back of the Cube
update the Counter
End_for

For all sensor readings in the Buffer do
generate 2-itemsets between the sensor readings in the particular round
add new nodes at the front of the Cube
discard the oldest nodes at the back of the Cube
update the Counter
End_For
estimateValue()
For all missing values do
estimate the missing value
store it in the Buffer
End_For
update()

```

Note 1: The Cube keep track of all existing 1- and 2-itemsets in each round, which are stored in the corresponding nodes and slices.

Note 2: The Counter data structure speeds up the estimation of a missing value.

Slika 22. Pseudokod za distribuirani FARM algoritam.

Slično rešenje, opisano u radu pod referencom [Boukerche, 2007] sa nazivom KDP (skraćeno od *Knowledge Discovery Process*, Proces pronalaženja znanja, istraživanje vršeno na Univerzitetu u Otavi, Kanada). Osnovna razlika je u tome

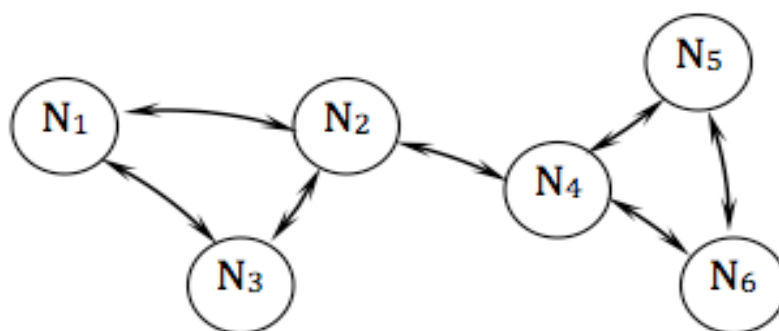
što se kod KDP algoritma senzorska mreža prvo deli u grupe, metodom za grupisanje čvorova, pri čemu se za svaku grupu određuje i čvor koji je vođa grupe. Metoda grupisanja se bazira na pokrivanju delova oblasti u kojoj se nalazi senzorska mreža, tako da se čvorovi koji pokrivaju jedan sektor nalaze u istoj grupi. Svaki čvor u jednoj grupi treba da pokriva ceo sektor za koji je data grupa vezana, i pretpostavlja se da su očitavanja sa senzora na čvoru dominantno definisana događajima koji se dešavaju u tom sektoru. Ukoliko su ovi uslovi ispunjeni, očitavanja sa senzora na čvorovima koji su istoj grupi će biti visoko korelisani. Istovremeno, pošto jedna grupa čini samo mali podskup cele senzorske mreže, asocijativna pravila je moguće pronaći uz znatno manje računarskih resursa. Iz ovog razloga je moguće celokupno izračunavanje smestiti unutar bežične senzorske mreže, odnosno na pojedinim čvorovima (konkretno, na čvorovima koji su označeni kao vođe grupa).

2.3.4.2. Pronalaženja asocijativnih pravila sa horizontalno distribuiranim podacima

Algoritam za pronalaženje asocijativnih pravila u horizontalno distribuiranim podacima u senzorskim mrežama je opisan u radu pod referencom [Roemer, 2007] (istraživanje obavljeno na ETH, Cirihi). Pošto algoritmu nije dat naziv, označićemo ga kao Romer algoritam, prema prvom autoru rada. U toku pripreme za rad algoritma svaki čvor izvršava lokalnu proceduru kojom pronalazi korelacije između očitavanja sa senzora koji se nalaze na njemu i događaja u okruženju. Događaju su definisani kao broj senzorskih susednih čvorova (čvorova koji se nalaze u fizičkom okruženju datog čvora) koji detektuje neko dešavanje u okolini.

Zatim, u toku rada algoritma, očitavanja sa senzora se diskretizuju u definisan broj klasa (na primer visoka, umerena, ili niska vrednost). Diskretizacija se vrši i po vremenskoj osi, odnosno očitavanja se svrstavaju u klase po trenutku kad su uzeta u odnosu na sadašnjost (na primer skorašnje, davna prošlost, itd.), kao i po procenjenoj fizičkoj udaljenosti posmatranog dešavanja (na primer blisko,

udaljeno, itd.). Za svaki set očitavanja ubacuje se po jedan red u tabeli u formatu $I(e, d, t, n)$, za sve vrednosti e, d, t, i i n parametara. Parametar e ovde označava tip događaja koji je detektovan, d označava odstojanje od čvora, t označava vremenski slot oko očitavanja. Parametar n označava broj senzorskih čvorova koji su detektovali događaj, koji se, u vremenskom slotu t , dogodio na odstojanju d od datog senzorskog čvora. Na primer, $I(„eksplozija”, blizu, skoro, 3)$ označava da su 3 čvora detektovala eksploziju u blizini datog čvora u kratkom vremenskom intervalu oko trenutka nastanka instance I . Rezultujuća tabela je ulaz za algoritam koji pronalazi asocijativna pravila. Na Slici 23 je prikazan tok podataka u mreži u toku rada distribuiranog Romer algoritma, dok Slika 24 daje implementaciju algoritma u pseudokodu.



Slika 23. Obrazac komunikacije u bežičnoj senzorskoj mreži u toku rada distribuiranog Romer algoritma.

The Roemer approach algorithm
For_each data point do Discretize readings Poll all neighboring nodes regarding detected events Create table entry $I(e, d, t, n)$ End_For Run a traditional Association Rule Mining algorithm, such as the Apriori over the construted table
e - type of event, spatial distance from the node taken into account, t - timeslot around the reading that is taken into account, n number of sensor nodes that report occurrence of the event type e in the time slot t , which are at most at the distance d from the node

Slika 24. Pseudokod za distribuirani Romer algoritam.

Iz ugla pojedinačnih senzorskih čvorova Romer algoritam se može posmatrati kao metoda nadgledanog mašinskog učenja, pošto se toku rada koriste informacije o tome da li se neki događaj dogodio ili nije. Međutim, iz ugla rada

cele mreže, algoritam je nenadgledan, pošto sve informacije nastaju u samoj mreži, u toku rada.

2.4. Diskusija postojećih rešenja

Po datom opisu (odabranih) postojećih rešenja iz otvorene literature, u tekstu koji sledi je data analiza i diskusija karakteristika tih rešenja. Fokusiraćemo se na nekoliko elemenata koji su važni pri tipičnim upotrebama bežičnih senzorskih mreža. Konkretno, biće uzeti u obzir:

- Sposobnost algoritma da radi sa mobilnim sensorima i promenama u okruženju;
- Sposobnost algoritma da reši problem podataka koji nedostaju;
- Očuvanje privatnosti u toku rada algoritma;
- Zahtevnost u pogledu komunikacije u toku rada algoritma;
- Kašnjenja koja unosi algoritam pri radu.

Svrha ovog poglavlja nije da se ponovo razmatra svako od navedenih rešenja pojedinačno, već da dâ generalizovanu diskusiju i izvuče zaključke u vezi sa različitim pristupima i rešenjima koji su korišćeni u razvoju raznih algoritama za distribuirano mašinsko učenje u bežičnim senzorskim mrežama.

2.4.1 Rad sa mobilnim sensorima i promenama u okruženju

Mnoge primene bežičnih senzorskih mreža zahtevaju upotrebu mobilnih senzorskih čvorova, odnosno čvorova čija se lokacija menja u vremenu. Posmatrano iz ugla mašinskog učenja, pokretni čvorovi mogu dovesti do promena u korelacijama među podacima koji se očitavaju sa senzora (a u slučaju nadgledanog učenja i u korelacijama između podataka očitanih sa senzora i vrednosti labela). Promene se mogu dogoditi i među lokalnim podacima (podacima koji nastaju i koji se nalaze na jednom čvoru), kao i među podacima sa različitih čvorova. Efekat je takav da čim se bar jedan čvor u mreži pomeri,

može doći do toga da modeli koji su trenirani na podacima nastalim pre pomeranja ne budu više validni.

Nešto drugačiji problem, sa istim efektima, nastaje ukoliko dođe do promena u okruženju, koje sa sobom povlače promene u korelacijama među podacima očitavanim sa senzora. Kao primer, posmatrajmo bežičnu senzorsku mrežu koja služi za nadgledanje pošumljenog prostora, sa ciljem otkrivanja šumskih požara. Ukoliko se posmatraju očitavanja u toku noći, već i malo povećanje temperature može označiti vatru u blizini senzorskog čvora. Sa druge strane, ista ta vrednost u toku dana može biti sasvim regularno očitavanje, pogotovo ukoliko se senzorski čvor nalazi na osunčanoj lokaciji. Dakle, mobilni čvorovi i izmene u okruženju mogu izazvati promene u zakonitostima u podacima i njihovom značenju.

Neka od rešenja za mašinsko učenje u bežičnim senzorskim mrežama imaju sposobnost da ispravno funkcionišu i u situacijama kada je ovaj problem prisutan (Tabela 1). U opisanim rešenjima mogu se uočiti dva osnovna principa kako se to ispravno funkcionisanje postiže:

1. Sistemi koji primenjuju takozvano *aktivno (online)* učenje imaju „urođenu“ sposobnost da reše probleme vezane za mobilne čvorove i promene u okruženju. Pojam aktivnog učenja podrazumeva da ne postoje odvojene faze, prva u kojoj se isključivo vrši analiza podataka i kreiranje modela i druga u kojoj se kreirani modeli isključivo koriste za predviđanje. Umesto toga, sistemi sa aktivnim učenjem konstantno osvežavaju modele na osnovu novih podataka koji pristižu, i koji se koriste i za predviđanja. Time ovakvi sistemi prirodno evoluiraju zajedno sa korelacijama u podacima sa kojima se koriste.
2. Sistemi koji primenjuju *neaktivno (offline)* učenje (sa odvojenim fazama za učenje i predviđanje) mogu biti prošireni okidačem koji pokreće ponovno pravljenje modela, odnosno vraća sistem u fazu učenja, ovaj put sa svežim podacima. Okidači se najčešće vezuju za pad performansi modela, bilo na

pojedinačnim čvorovima, bilo u celokupnom sistemu, dok se kao performanse posmatraju tačnost, prosečna greška, preciznost, itd. Ukoliko ova metrika padne ispod predefinisane vrednosti, okidač se aktivira.

2.4.2. Vrednosti koje nedostaju

Problem sa vrednostima koje nedostaju je dobro poznat u domenu mašinskog učenja. Do njega dolazi kada su vrednosti nekih atributa (u primenama vezanim za bežične senzorske mreže jedan atribut predstavlja jedno očitavanje sa nekog senzora u mreži) nepoznate za neke od instanci podataka u skupu. Problem se može javiti i u toku učenja i u toku primene naučenog modela.

Sa druge strane, bežične senzorske mreže su visoko podložne kvarovima i greškama u komunikaciji, usled kojih se može desiti da podaci ne stignu na odredište, ili stignu u neispravnom obliku. Štaviše, u mnogim sistemima bežičnih senzorskih mreža, čvorovi se namerno isključuju u određenim trenucima, kako bi se sačuvala energija u baterijama. U svakom od ovih slučajeva, rezultat iz ugla mašinskog učenja je nedostajući podatak. Dakle, kod rada sa bežičnim senzorskim mrežama se može očekivati da će se veoma često javljati situacija u kojoj neki podaci nedostaju.

Tipičan način rešavanja problema koji nastaju usled kvarova, grešaka u komunikaciji i ugašenih čvorova je oslanjanje na visok nivo redundantnosti. Tačke u oblasti koju pokriva jedna senzorska mreža su često pokrivene sensorima sa nekoliko senzorskih čvorova. Čak iako podaci sa jednog ili čak većeg broja čvorova nisu dostupni, uglavnom će postojati podaci sa nekog drugog čvora koji pokriva istu tačku. Posmatrano kroz prizmu teorije informacija i mašinskog učenja to znači da je većina informacija sadržana u podskupovima celokupnog skupa podataka, te je do optimalnih odluka moguće doći posmatranjem samo nekog od tih podskupova.

Tradicionalno, postoje tri načina kojima se pri mašinskom učenju rešavaju situacije u kojima nedostaju neke od vrednosti u skupu podataka koji se analizira:

1. Brisanje svih instanci podataka kod kojih neke vrednosti nedostaju. Ovakav pristup nije pogodan kod primena u bežičnim senzorskim mrežama, pošto bi veliki deo podataka morao da bude obrisano.
2. Upotreba algoritama koji imaju ugrađenu sposobnost da rade sa skupovima podataka u kojima su neke od vrednosti nepoznate (jedan primer je većinska logika, gde se na osnovu svakog poznatog podatka nezavisno donosi predlog odluke, pa se ti predlozi prebrojavaju; kod podataka čije vrednosti nedostaju se predlozi jednostavno ne prave).
3. Upotreba tehnika pretprocesiranja podataka, kojom se nedostajuće vrednosti popunjavaju na neki način. Na primer, može se koristiti neka predefinisana vrednost, uzeti prosečna ili najčešća vrednost date promenljive u skupu podataka, ili pokušati sa procenom na osnovu drugih promenljivih.

U sklopu primena mašinskog učenja u bežičnim senzorskim mrežama problem nepoznatih vrednosti se razlikuje u zavisnosti od načina na koji se posmatra distribucija podataka u mreži. Radi jednostavnosti u diskusiji koja sledi pretpostavićemo da situacija kada neke vrednosti nedostaju nastaje tako što izostanu sva očitavanja koja bi trebalo da stignu sa nekog udaljenog čvora (bilo da se radi o analizi na centralnoj lokaciji van mreže, ili na nekom čvoru unutar mreže, u šemi u kojoj čvorovi u toku rada razmenjuju očitavanja sa senzora).

1. Ukoliko se radi o sistemu u kome je distribucija podataka horizontalna, nepoznata vrednost ima efekat izbacivanja cele instance iz skupa podataka. Ovakva situacija ne predstavlja ozbiljan problem pri mašinskom učenju, osim što smanjuje količinu podataka koji su na raspolaganju. Kako se pri horizontalnoj distribuciji lako generišu jako velike količine podataka, takvo smanjenje nema veliki uticaj. Međutim, prilikom razvoja algoritma treba voditi računa da se sistem ne blokira ukoliko do gubitka podataka zaista

dode, na primer, ukoliko postoji sinhronizacija na nekoj barijeri, ili neki od elemenata u sistemu čeka da stignu podaci sa svih čvorova. Takođe, ipak treba napomenuti da u slučaju da je procenat vrednosti koje nedostaju veoma visok, može doći do poremećaja u distribucijama podataka, pri čemu se korelacije menjaju, pa naučeni modeli mogu postati neispravni.

2. Ukoliko se radi o sistemu sa vertikalnom distribucijom podataka, onda će nepoznate vrednosti stvoriti nepotpune instance podataka u globalnom skupu podataka. Ovakva situacija je komplikovanija za rešavanja i analogna problemu nepoznatih vrednosti kod klasičnog mašinskog učenja. U nekim opisanim rešenjima je ovakav problem rešavan baš na načine koji su navedeni kao rešenja kod tradicionalnog mašinskog učenja (za više detalja videti Tabelu 1).

2.4.3. Očuvanje privatnosti

U kontekstu bežičnih senzorskih mreža očuvanje privatnosti se odnosi da izdavanje informacija koje su od interesa za korisnike bez eksplicitnog otkrivanja vrednosti koje su očitane sa senzora, ukoliko to nije eksplicitno neophodno. Očuvanje privatnosti se može garantovati na dva nivoa:

1. Očuvanje privatnosti na nivou individualnih senzorskih čvorova je stroži skup zahteva, i podrazumeva da očitavanja sa senzora nikada ne napuštaju sam čvor na kojem su nastala. Upotrebom mašinskog učenja postoje dva načina kako se ovakvi zahtevi mogu ispuniti. Prva opcija je pravljenje lokalnih modela metodama mašinskog učenja, na osnovu lokalnih podataka, i deljenje samo izlazi iz ovakvih lokalnih modela. Drugi način podrazumeva pravljenje jedinstvenog globalnog modela nekom kooperativnom metodom mašinskog učenja, tako da se sam model prenosi kroz mrežu, dok očitani podaci ostaju na čvorovima.
2. Očuvanje privatnosti na nivou mreže uključuje relaksiraniji skup zahteva i podrazumeva da podaci očitani sa senzora ne napuštaju granice senzorske mreže, dok se mogu razmenjivati između čvorova unutar mreže. Ukoliko

pretpostavimo postojanje neke centralne lokacije, koja ima veći kapacitet od čvorova unutar mreže, slanje podataka na ovakvu lokaciju bi narušilo očuvanje privatnosti na nivou mreže.

2.4.4. Zahtevnost u pogledu komunikacije

Kao što je već bilo reči, bežične senzorske mreže su veoma ograničene u pogledu kapaciteta kada je reč o komunikaciji među čvorovima. Jedan od osnovnih razloga za upotrebu distribuiranog mašinskog učenja u ovakvim sistemima je upravo mogućnost smanjivanja količine komunikacije u toku rada. Sa stanovišta neophodne komunikacije kod algoritama distribuiranog mašinskog učenja, veoma bitan faktor je da li se primenjuje aktivno učenje (model se istovremeno kreira i koristi u toku rada) ili neaktivno učenje, kod kojega prvo postoji faza treniranja modela, pa se potom odvija faza korišćenja.

Kod pristupa sa neaktivnim učenjem obrasci komunikacije su najčešće različiti u fazama treniranja i korišćenja modela. Faza treniranja se odvija samo jednom (ili više puta, ukoliko je u pitanju sistem sa okidačem za ponovno treniranje), dok čvorovi provode veći deo vremena u fazi korišćenja modela. Ipak, faza treniranja može dovesti do relativno velike potrošnje energije u sistemu, ukoliko su zahtevi za komunikacijom veoma visoki. U pogledu obrasca komunikacije u fazi treniranja izdvaja se nekoliko kategorija rešenja:

1. Treniranje lokalnih klasifikatora samo na osnovu podataka koji su dostupni na individualnim čvorovima. Pristupi sa ovakvim treniranjem zahtevaju najmanju količinu komunikacije (samo ukoliko se koristi nadgledano učenje potrebno je preneti informaciju o vrednostima labela do čvora).
2. Inkrementalno kooperativno treniranje globalnog modela, sa jednim prolazom kroz mrežu. Kod ovakvih pristupa čvor prihvata trenutnu verziju modela, po potrebi je modifikuje, i šalje do sledećeg čvora. Model prolazi kroz celu mrežu, uz samo jednu posetu svakom čvoru. U ovom slučaju komunikacija zavisi od veličine (i topologije) mreže, kao reprezentacije

modela koja se šalje, odnosno od mogućnosti da se model efikasno kodira sa što manjim brojem bita.

3. Razmena podataka između međusobno bliskih čvorova podrazumeva komunikaciju u okviru malih razdaljina, najčešće tako da čvorovi mogu direktno da komuniciraju. Kod ovakvih pristupa neophodna količina komunikacije u mnogome zavisi od topologije mreže (broja bliskih čvorova sa kojima se razmenjuju podaci) i količine podataka koja se razmenjuje. Pošto se izbegava komunikacija između udaljenih čvorova koja zahteva veoma velike količine energije, ovakav tip treniranja se može smatrati kao relativno efikasan, iako može zahtevati razmenu velikih količina podataka.
4. Iterativno treniranje, kod kog se ili model ili neki lokalni podaci primaju i šalju sa svakog čvora veći broj puta. Da bi pristupi sa ovakvom fazom treniranja bili energetske efikasni, neophodno je da budu upareni sa mehanizmom rada u fazi korišćenja koja troši veoma malo energije. Takođe, potrebno je da aplikacija zadovoljava uslov da se treniranje vrši jednom (ili vrlo malo puta), a da faza korišćenja traje dugo. Drugim rečima, ovakvo treniranje može biti pogodno samo ukoliko modeli napravljeni mašinskim učenjem ostaju validni duži vremenski period (nema velikih promena u okruženju i senzorski čvorovi su statični).

U toku korišćenja neaktivno napravljenih modela, kao i sistema sa aktivnim učenjem, u pogledu komunikacije moguća su tri osnovna principa rada.

1. Donošenje lokalnih odluka, na osnovu lokalnih modela i lokalnih podataka (pojam lokalno podrazumeva da se svi elementi nalaze na individualnim čvorovima, bez komunikacije). Ovakav pristup je karakterističan za klasifikacione sisteme: lokalni modeli se koriste kako bi se napravile lokalne odluke, koje se onda šalju kroz mrežu i agregiraju u globalnu odluku. Na ovaj način se postiže ušteda pri komunikaciji, pošto je dovoljno slati klase labele, koje se mogu kodirati sa malim brojem bita, umesto očitavanja sa senzora, koja mogu biti velika.

Iz ugla teorije informacija, ovi pristupi se mogu posmatrati kao vrsta redukcije dimenzija problema, kada se informacija od interesa (vrednost labele) izvlači iz veće reprezentacije (očitavanja sa senzora) i kodiraju u manju reprezentaciju (reprezentacija vrednosti labele).

2. Približno određivanje vrednosti koje su očitane na nekim udaljenim čvorovima, što se može vršiti na drugim čvorovima unutar mreže ili na nekoj centralnoj lokaciji. Tipično za pristupe koji se baziraju na regresiji i pronalaženju asocijativnih pravila, ovo rešenje omogućava senzorskim čvorovima da se uzdrže od komunikacije, pošto se vrednosti koje su očitane sa njihovih senzora mogu približno odrediti na drugim mestima. Kako pri procenama može doći do greške, često se na svakom čvoru vrši i procena lokalnih parametara i komunikacija pokreće ukoliko je greška prevelika (iznad neke utvrđene vrednosti).
3. Donošenje globalne odluke na osnovu podataka koji su prikupljeni iz nekog podskupa svih čvorova u mreži. Osnovna ideja ovakvog pristupa je slična prethodno opisanom rešenju, uz razliku da je umesto određivanja približnih vrednosti očitavanja sa senzora udaljih čvorova direktno pokušavaju odrediti informacije koje su od interesa za korisnika. Nekoliko opisanih algoritama grupisanja i pronalaženja asocijativnih pravila prati ovakav pristup. Rešenja je moguće dalje proširiti tako da uključuju efikasne tehnike balansiranja opterećenja pojedinih senzorskih čvorova, i tako dodatno produžiti životni vek mreže.

Drugi i treći mehanizam rada algoritama za mašinsko učenje u bežičnim senzorskim mrežama, koji su iznad opisani, baziraju se na ideji da korelacije između očitavanja sa senzora na različitim čvorovima u datom trenutku predstavljaju posledicu redundantnosti u informacijama koje nastaju u ovakvim mrežama. Treba obratiti pažnju na činjenicu da se ova redundantnost upotrebljava i kod nekih rešenja za problem nepoznatih vrednosti. Iz ovog

razloga treba paziti kod kombinovanja ovakvih tehnika, pošto se može dogoditi da postoji dovoljno redundantnih informacija za primenu jednog, ali ne oba mehanizma.

Konačno, treba uzeti u razmatranje i uticaj distribucije podataka u mreži koju pretpostavlja neko rešenje na mogućnosti za smanjivanje komunikacije u mreži prilikom treniranja i korišćenja sistema.

Ukoliko algoritam pretpostavlja da su podaci horizontalno distribuirani, to povlači da cele instance podataka nastaju na individualnim čvorovima. To znači da se odluke mogu donositi nezavisno na svakom čvoru, bez ikakve komunikacije, ukoliko na datom čvoru postoji kopija modela koji je napravljen. U fazi pravljenja modela komunikaciji se pribegava kako bi se povećali skupovi podataka ili u obzir uzele i korelacije koje su vidljive kroz podatke koji se trenutno nalaze na drugim čvorovima, ali suštinski komunikacija nije neophodna.

Ukoliko algoritam pretpostavlja da su podaci vertikalno distribuirani, jedna instanca podataka se sastoji od očitavanja koja nastaju na većem broju (ili svim) čvorova u mreži. To znači da je izvesna količina komunikacije neophodna ako se odluke žele donositi uz upotrebu svih atributa u jednoj instanci.

2.4.5. Kašnjenja pri radu sistema

Poslednji aspekt algoritama za mašinsko učenje prilagođenih radu u bežičnim senzorskim mrežama koji će biti razmatran se odnosi na kašnjenje pri radu sistema (odnosno u fazi korišćenja modela, ukoliko se radi o sistemu sa neaktivnim učenjem). Pod pojmom kašnjenje podrazumevamo vremenski interval koji prođe od trenutka kada su podaci očitani sa senzora na pojedinim čvorovima, do trenutka kada je globalna odluka poznata. U mnogim primenama bežičnih senzorskim mreža kašnjenje pri radu je veoma visokog značaja [Li, 2007]. U poređenju sa standardnim načinom upotrebe bežičnih

senzorskih mreža, gde se podaci prvo šalju i prikupljaju na jednoj lokaciji, pa potom obrađuju, distribuirana rešenja koja su opisana u ovoj tezi donose značajne mogućnosti za smanjivanje kašnjenja u radu. Pri analizi koja sledi, kod sistema sa neaktivnim učenjem, u obzir će biti uzeto samo kašnjenje koje nastaje u fazi korišćenja sistema, dok se ignoriše faza treniranja. Razlog nalazimo u tome što se može smatrati da je sistem funkcionalan tek u trenutku kada je treniranje završeno.

Kod bežične senzorske mreže mogu se definisati dva izvora kašnjenja: kašnjenje koje nastaje usled komunikacije (vreme koje protekne dok se podaci pripremaju za prenos i prenose) i kašnjenje usled izračunavanja (vreme koje protekne u izvršavanju instrukcija kojima se model primenjuje na podatke). Pokazuje se da kašnjenje usled komunikacije znatno veće od kašnjenja pri izračunavanju, čak i u situacijama gde su izračunavanja relativno kompleksna, a prenosi se mala količina podataka. Iz ovog razloga fokusiraćemo se na analizu kašnjenja koja nastaju usled komunikacije.

Pri radu opisanih algoritama mogu se izdvojiti tri slučaja:

1. Algoritmi u kojima se lokalne odluke donose na čvorovima, uz upotrebu samo lokalnih podataka, a potom šalju na centralnu lokaciju gde se spajaju u globalnu odluku. Kod ovih algoritama nema značajnog smanjenja u kašnjenju u odnosu na klasična rešenja (osim što se deo procesiranja izvršava distribuirano i u paraleli).
2. Algoritmi kod kojih se globalne odluke donose na čvorovima, posle izvesne količine komunikacije sa čvorovima koji se nalaze u neposrednoj blizini. Ovakvi algoritmi omogućavaju izvesno smanjenje kašnjenja pri radu, pošto se komunikacija odvija samo između čvorova koji mogu direktno da komuniciraju.

3. Algoritmi kod kojih se sve odluke donose lokalno na individualnim čvorovima, bez ikakve potrebe za komunikacijom. Kod ovakvih rešenja se kašnjenja pri radu mogu značajno smanjiti.

Tabela 1 daje pregled algoritama iz otvorene literature koji su razmatrani u prethodnom tekstu, zajedno sa rešenjima i karakteristikama vezanim za navedenih pet problema koji su bitni za upotrebu.

Tabela 1: Osobine analiziranih rešenja iz otvorene literature. Tipovi: K – klasifikacija, R – Regresija, G – Grupisanje, A – Pronalaženja asocijativnih pravila

Algoritam	Tip	Distribucija podataka	Mobilni čvorovi, promene u sredini	Nepoznate vrednosti podataka	Privatnost	Potreba za komunikacijom	Kašnjenje	Osnovna namena
Mekonel	K	Vertikalna	Da; Okidač za učenje	Da; Većinska logika	Na čvoru	Niska	Visoko	Detekcija događaja
distMAS C	K	Vertikalna	Da; Aktivno učenje	Ne	Ne	Visoka	Visoko	Detekcija događaja
DFP-SVM	K	Vertikalna	Da; Okidač za učenje	Ne – pri učenju Da – pri radu; lokalne odluke	U mreži	Umerena	Nisko	Detekcija događaja
SSG-SVM	K	Vertikalna	Da; Okidač za učenje	Da; Većinska logika	U mreži	Umerena	Umereno	Detekcija događaja
SPIRIT	R	Vertikalna	Da; Aktivno učenje	Ne	Ne	Visoka	Visoko	Detekcija događaja
PREMON	R	Horizontalna	Da; Aktivno učenje	Da; Procena vrednosti	Ne	Umerena	Nisko	Smanjanje potrošnje
TREG	R	Horizontalna	Ne	Da; Procena vrednosti	U mreži.	Niska	Nisko	Smanjanje potrošnje
KREG	R	Horizontalna	Ne	Ne – pri učenju Da – pri radu; lokalne odluke	Na čvoru	Niska	Nisko	Detekcija događaja
RO-KREG	R	Horizontalna	Ne	Ne – pri učenju Da – pri radu; lokalne odluke	U mreži	Umerena	Nisko	Detekcija događaja
DHC	G	Vertikalna	Ne	Ne	Ne	Visoka	Visoko	Otkrivanje zakonitosti
K-means	G	Horizontalna	Ne	Da; Većinska logika	Na čvoru	Niska	Nisko	Otkrivanje zakonitosti
FARM	A	Vertikalna	Da; Aktivno učenje	Da; Procena vrednosti	Ne	Visoka	Nisko	Rešavanje podataka koji nedostaju
KDP	A	Vertikalna	Da;	Da;	U mreži	Umerena	Nisko	Rešavanje

			Aktivno učenje	Procena vrednosti				podataka koji nedostaju
Roemer	A	Horizontalna	Ne	Da; Većinska logika	Na čvoru	Niska	Nisko	Detekcija događaja

2.4.6. Zaključci na osnovu analize postojećih rešenja

Na osnovu analize sadržaja datog u Tabeli 1 moguće je doneti sledeći važan zaključak: ponašanje algoritma u pogledu pet karakteristika (5.2 - 5.7) koje su bitne kod primene mašinskog učenja u bežičnim senzorskim mrežama u velikoj meri zavisi od obrazaca komunikacije pri radu sistema u toku izvršavanja datog algoritma. Iako se opisani algoritmi dosta međusobno razlikuju, moguće je uočiti nekoliko osnovnih generičkih tipova rešenja. U nastavku je data diskusija ovih tipova i ponašanja algoritama u pogledu navedenih pet karakteristika. Isti zaključci su dati i u Tabeli 2.

1. *Rešenja koja se baziraju na lokalnom treniranju modela pri čemu se koriste samo lokalni podaci, praćenom slanjem odluka iz lokalnih modela do centralne lokacije, gde se na osnovu njih donosi globalna odluka.*

U fazi treniranja ovakvi algoritmi ne zahtevaju nikakvu komunikaciju, pa su prema tome veoma energetski efikasni. U fazi rada se odluke šalju do udaljene lokacije, što može zahtevati i višestruko primanje i slanje paketa (multi-hop komunikacija). Međutim, ukoliko se podaci koji se šalju mogu efikasno kodirati malim brojem bitova, i ukoliko se koristi agregacija podataka na putanji, algoritmi sa ovim pristupom mogu biti relativno energetski efikasni, čak i u fazi korišćenja modela.

Pošto se odluke iz cele mreže slivaju u jednu, centralnu tačku, algoritam teorijski može da pronađe zakonitosti koje obuhvataju informacije koje su razbacane po celoj mreži. Sa druge strane, ovi pristupi zahtevaju da su lokalni skupovi podataka dovoljno informativni, da bi se na osnovu njih mogle

napraviti lokalne odluke, bez (velikih) gubitaka u informacijama. Ukoliko donošenje odluke zahteva istovremeno poznavanje vrednosti koje se očitavaju sa senzora koji se nalaze na različitim čvorovima, ovakvi pristupi neće raditi (jedan primer, o kojem će biti više reči u drugom delu ove teze je poznati XOR problem [Jakulin, 2004]).

Može se očekivati da vreme koje će mreža provesti u fazi treniranja bude relativno kratko, jer nema komunikacije. Kašnjenja u toku rada u velikoj meri zavise od topologije mreže i mogu biti veoma velika, ukoliko postoje duge putanje sa višestrukim slanjem i primanjem podataka.

2. Rešenja koja se baziraju na komunikaciji isključivo između međusobno bliskih čvorova u mreži

Kod ovih rešenja komunikacija postoji samo između čvorova koji su međusobno fizički bliski, odnosno koji mogu direktno da komuniciraju. Komunikacija može da se odvija u fazi treniranja, fazi korišćenja modela, ili u obe faze. U nekim rešenjima se razmenjuju podaci koji su očitani sa senzora, a u nekima modeli koji si izgrađeni od lokalnih podataka. Komunikacija može biti neophodna samo jednom (u fazi treniranja, ili po jednom za svaku odluku u toku korišćenja modela) ili iterativno veći broj puta (u fazi treniranja ovo bi značilo da se podaci ili model prenose više puta i u svakom prolazu modifikuju, dok u bi u fazi korišćenja značilo da se odluke donose iterativno, što je veoma retko).

Iako kod algoritama koji prate ovakav metod rada postoji određena količina komunikacije, ona je uvek ograničena na bliske čvorove koji mogu direktno da komuniciraju, čime se izbegava energetska skup *multi-hop* komunikacija. Iz ovog razloga dati algoritmi mogu da budu relativno efikasni u pogledu potrošnje energije, pogotovo ukoliko nisu u pitanju iterativne procedure.

Korelacije koje je moguće pronaći i iskoristiti u ovakvim modelima uključuju one koje su vidljive u očitavanjima sa senzora na pojedinačnim čvorovima. Takođe, mogu se pronaći i one zakonitosti koje zahtevaju istovremeno poznavanje vrednosti koje se očitavaju sa senzora koji se nalaze na različitim čvorovima (pomenuti XOR problem), ali samo među podacima sa međusobno bliskih senzora. Međutim, zakonitosti koje su uočljive na osnovu podataka sa udaljenih delova mreže je veoma teško (ako ne i nemoguće) pronaći.

Vreme treniranja algoritama koji prate ovakav pristup u velikoj meri zavisi od izbora da li se modeli kreiraju iterativno ili ne (pri čemu prvi slučaj može trajati znatno duže), i u obe varijante se može očekivati da bude umereno do relativno dugo (u poređenju sa algoritmima koji se treniraju sa lokalnim podacima). Što se tiče kašnjenja u radu, može se očekivati da ona budu među najmanjima među rešenjima koja su opisana u ovom tekstu, s obzirom na to da nema dugih multi-hop komunikacija. Nivo očuvanja privatnosti zavisi od toga da li se razmenjuju podaci očitani sa senzora ili lokalni modeli. U prvom slučaju obezbeđeno je očuvanje privatnosti na nivou mreže, dok drugi slučaj garantuje očuvanje privatnosti na nivou senzorskih čvorova. Problemi sa vrednostima koje nedostaju se moraju rešavati na nivou čvorova.

3. Rešenja kod kojih se globalni modeli grade sekvencijalnim prolazima kroz celu mrežu

U algoritmima sa ovim tipom pristupa grade jedan jedinstven model, tako što svaki od čvorova dobije kopiju trenutnog modela, po potrebi je izmeni na osnovu svojih lokalnih podataka i prosledi dalje do sledećeg čvora. U fazi korišćenja modela, odluke se najčešće donose lokalno na pojedinim čvorovima (mada su moguća i proširenja koja uključuju izvesnu komunikaciju). Procedura izgradnje modela može uključivati jedan prolazak kroz celu mrežu, ili iterativnu proceduru sa većim brojem prolaza.

U toku treniranja modela, rešenja koja prate ovakav pristup zahtevaju veliku količinu komunikacije, pogotovo ukoliko se prati iterativni pristup. Sa druge strane, u toku rada sistema, odnosno korišćenja modela, komunikacije ili nema uopšte ili je veoma limitirana, što ovakve algoritme čine energetski efikasnim u toku tog perioda.

Algoritmi koji pripadaju ovoj grupi mogu da pronađu zakonitosti koje uključuju podatke iz udaljenih delova mreže, pošto jedan model obilazi celu mrežu u toku treniranja. U zavisnosti od implementacije, u nekim slučajevima je moguće pronaći i zakonitosti koje su vidljive samo ukoliko se istovremeno posmatraju podaci koji se nalaze na različitim čvorovima (pomenuti XOR problem).

Vreme koje sistemi sa algoritmima iz ove grupe rešenja provedu u treniranju modela je relativno veliko, pošto treniranje zahteva sekvencijalno prolaženje kroz celu mrežu, uz primanje i slanje modela do svakog čvora. Pored ove komunikacije, i količina paralelizma koji se može iskoristiti je ograničena, zato što svaki čvor čeka da dobije jedinstveni model i tek onda izvrši obradu. Kašnjenje u toku rada je, sa druge strane, veoma malo, pošto se odluke najčešće donose lokalno, bez komunikacije (osim ako se ne uvede modifikacija koja zahteva izvesnu količinu komunikacije, sa ciljem poboljšanja performansi). Rešenja iz ove grupe garantuju privatnost na nivou čvorova. Lokalne odluke čine ovakva rešenja robusnim u pogledu vrednosti koje nedostaju.

Tabela 2: Pregled tipova rešenja na visokom nivou apstrakcije

	<u>Potreba za komunikacijom</u>	<u>Kašnjenje</u>	<u>Privatnost</u>	<u>Nepoznate vrednosti podataka</u>
Lokalno učenje, globalne odluke	Učenje - niska Rad - umerena	Učenje - nisko; Rad - umereno	U čvoru	Globalno rešenje
Komunikacija u	Učenje - umerena	Učenje - umereno	U mreži	Lokalno

okruženju	Rad - umerena	Rad - umereno		rešenje
Sekvencijalno prolaženje kroz mrežu	Učenje - umerena ili visoka Rad - niska	Učenje -visoko Rad - nisko	U čvoru	Lokalno rešenje

2.4.7. Performanse algoritama za distribuirano mašinsko učenje u bežičnim senzorskim mrežama

Veoma važno pitanje u vezi sa algoritmima za distribuirano mašinsko učenje u bežičnim senzorskim mrežama i njihovim razvojem se odnosi na to kako ih je najbolje evaluirati, jednom kada je algoritam razvijen. Autori različitih radova koji se pominju u ovom tekstu su pratili različite pristupe. Međutim, najčešći pristup u datim radovima je sledeći: očitavanja sa senzora se simuliraju tako što se vrednosti nasumično izvlače iz dve ili više Gausovih raspodela različitih srednjih vrednosti i varijansi. Jedan podatak se sastoji od M odabiraka iz jedne od distribucija (gde je M broj senzora u mreži). Prema tome, stanje u okruženju koje se želi otkriti na osnovu vrednosti očitanih sa senzora je identitet Gausove raspodele iz koje se „očitavanja“ izvlače.

Problem sa ovakvim načinom simuliranja podataka je u tome što u okruženju nema nikakvih prostornih efekata, odnosno veza između događaja ili stanja (trenutne Gausove raspodele) i očitavanja sa senzora (vrednosti izvučenih iz date raspodele) je invarijantna u odnosu na pozicije pojedinih senzorskih čvorova. Ukoliko bi posmatrali mrežu sa 100 čvorova, svaki sa po jednim sensorom, vektor podataka koji bi opisivao jednu instancu bi sadržao 100 nasumičnih odabiraka iz jedne Gausove raspodele. Ukoliko bi tu mrežu zamenili jednim čvorom koji može da generiše 100 očitavanja u veoma kratkom vremenskom intervalu (tako da se stanje okruženja ne može promeniti u toku tih 100 očitavanja) rezultujući skup podataka bi bio identičan. Iako je u nekim slučajevima ovakav način evaluacije opravdan, zaključci koji se mogu izvući su u najboljem slučaju ograničeni.

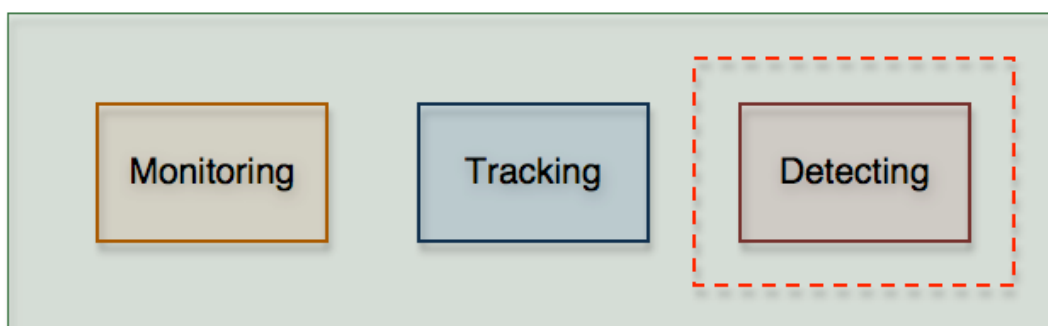
Kako bi se izbegao opisani problem, moguća su dva drugačija rešenja. Jedna mogućnost je ukoliko je akcenat na otkrivanju rezultata vezanih za teoriju informacija, odnosno tačnosti pri radu algoritma i njegovim sposobnostima da otkrije specifične tipove zakonitosti u podacima. U ovom slučaju se mogu koristiti dobro poznati skupovi podataka koji se koriste za evaluaciju algoritama mašinskog učenja (na primer *UCI Machine Learning Repository*, [Frank, 2010]), pri čemu se atributi „dodeljuju“ različitim sensorima, uz višestruko ponavljanje procedure evaluacije (na primer, primenom Monte Karlo metode). Drugi mogući pristup podrazumeva detaljniju simulaciju senzorske mreže, okruženja i vrednosti koje se mogu detektovati sensorima. U ovome slučaju se može razvijati simulacija na osnovu konkretne predviđene aplikacije sistema, sa željenim nivoom detalja. Osim toga što može dati procenu rada algoritma u uslovima koje diktira primena u konkretnoj aplikaciji, drugi opisani alternativni način evaluacije omogućava da se simulacija proširi kako bi se procenili i robustnost pri šumu i greškama u komunikaciji. Naravno, ovakav način evaluacije je ujedno i najkomplikovaniji.

Energetska efikasnost i smanjivanje potrošnje energije su jedan od osnovnih motiva za razvoj i upotrebu algoritama mašinskog učenja u bežičnim senzorskim mrežama. Kako je navedeno, u bežičnim senzorskim mrežama najveći deo energije se potroši pri komunikaciji. Da bi se postigle uštede u energiji, čest pristup je da se komunikacija izbegava kad god je to logički moguće, ostavljajući čvorove da komuniciraju samo kada je to apsolutno neophodno u algoritmu. Međutim, ovakvim pristupom se ne postižu uvek željeni rezultati. Razlog je u tom što su, osim strukturom algoritma, obrasci komunikacije u sistemu definisani i topologijom mreže, fizičkim rasporedom senzorskih čvorova, i strategijama rutiranja. Algoritamski je moguće eliminisati komunikaciju između obližnjih čvorova, ali ostaviti potrebu za komunikacijom između dva udaljena čvora. Ta komunikacija će se odvijati uz višestruke prijeme i slanja podataka, preko većeg broja čvorova. Koliko tačno čvorova će učestvovati je teško predvideti, pa je teško doneti odluke na nivou aplikativnog

algoritma (u ovom slučaju algoritma mašinskog učenja). Takođe, može se desiti da se u analizi podataka primenjuje neki mehanizam kojim se izbegava da se određeni podatak šalje do određenog čvora, ali da se isti podatak šalje do nekog trećeg čvora upravo preko čvora koji se pokušava izbeći, čime se ne pravi nikakva ušteda (podatak će svakako proći kroz dati čvor). Nešto drugačiji primer problema koji se može javiti se odnosi na sve šeme u kojima postoji neka vrsta grupisanja i odabira vođe grupe. Tu dolazi do neujednačenog opterećenja među čvorovima. Iako će se možda ukupna potrošnja energije u mreži smanjiti, čvorovi koji su više opterećeni će brže ostati bez napajanja, čime i cela mreža obično prestaje da radi (ili biva znatno oslabljena).

3. Vertikalno distribuirano mašinsko učenje i nove tehnike za određivanje težinskih faktora

Uobičajene aplikacije bežičnih senzorskih mreža se mogu podeliti prema osnovnom cilju koji se želi postići kao nadgledanje, praćenje ili detekcija, kao što je grafički prikazano na Slici 25. Istraživanje koje će u nastavku biti opisano se bavi senzorskim mrežama čija je osnovna svrha vezana za detekciju izvesnih događaja.

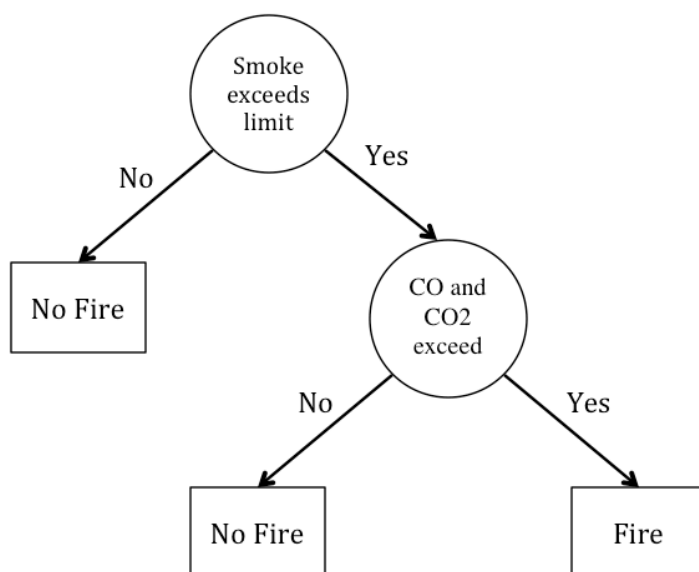


Slika 25: Aplikacije bežičnih senzorskih mreža podeljene prema osnovnom cilju koji se želi postići

Posmatrajmo sledeći scenario: jedna bežična senzorska mreža je postavljena u nekoj sredini. Svaki senzorski čvor je opremljen izvesnim brojem senzora pomoću kojih se dolazi do merenja nekih veličina u toj sredini, na osnovu kojih je moguće zaključiti da je došlo do neke velike promene. Promena može biti trajnog karaktera ili vremenski ograničena, jedino što je bitno je da izaziva uočljive promene u očitavanjima na (bar nekim) sensorima u našoj mreži.

Postavlja se pitanje na koji način najbolje obraditi podatke koje dobijamo iz ovakve mreže? Pretpostavimo da je mreža veoma velika i uključuje velike količine senzora. U takvom slučaju je ručna analiza podataka vrlo zahtevana, pa se može pristupiti upotrebi metoda mašinskog učenja. Kako se žele detektovati događaji koji se mogu opisati kao vrednosti iz definisanog konačnog skupa, problem se može posmatrati kao klasifikacioni problem. Klasifikator (ili klasifikacioni model) se gradi tako da uzima podatke očitane sa senzora i

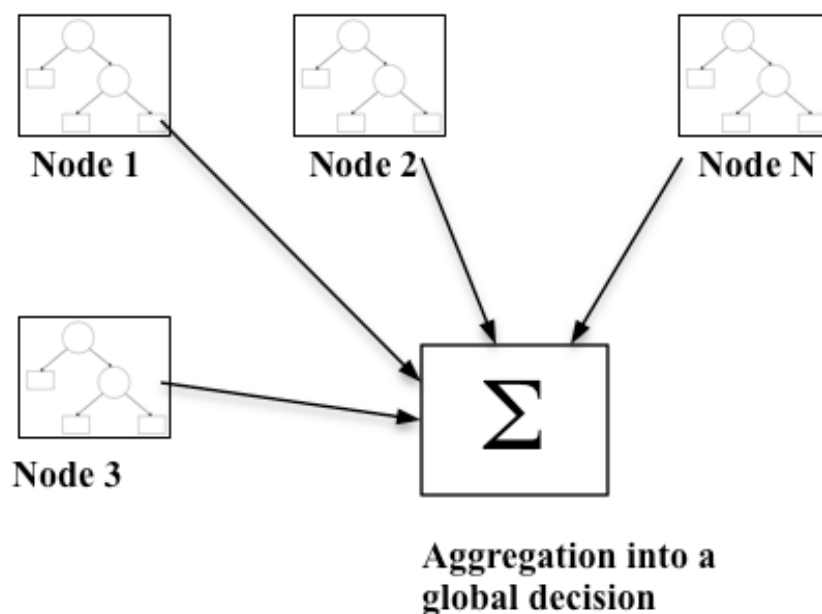
određuje izlaznu vrednost koja odgovara nekom od mogućih događaja koji se očekuju (ili vrednost koja označava da nema nikakvih događaja). Izlazna vrednost se naziva klasnom labelom. Primer klasifikatora čija je namena otkrivanje požara na osnovu senzora za temperaturu i dim je dat na Slici 26 [Chen, 2007]. Odluka o vrednosti izlazne promenljive se donosi tako što se vrši spuštanje niz stablo, pri čemu se uzima putanja koja odgovara vrednostima očitanim sa senzora.



Slika 26: Stablo odlučivanja za detekciju požara na osnovu vrednosti očitavanja senzora za temperaturu i dim.

U nastavku ćemo razmatrati mogućnost primene distribuiranog algoritma mašinskog učenja u ovakvoj mreži. Ideja je da se primeni algoritam koji pretpostavlja vertikalnu raspodelu podataka. Na svakom čvoru se izvršava deo algoritma mašinskog učenja, pri čemu se na svakom čvoru donose lokalne odluke. Ove odluke se potom na centralnoj lokaciji spajaju u globalnu odluku (Slika 27). Motivacija za primenu distribuiranog mašinskog učenja, umesto centralizovane analize podataka u kojoj bi se svi podaci prikupili na jednom mestu, pa tek onda obradili, jeste u smanjivanju neophodne komunikacije. To se postiže tako što se umesto očitavanja sa senzora šalju samo klasne labele, koje mogu imati mali broj mogućih vrednosti [McConnell, 2005]. Dodatna prednost distribuiranog mašinskog učenja je u tome što su vrednosti sa senzorskih

čvorova, koje se prikupljaju na centralnoj lokaciji, unificirani u strukturi i značenju. Drugim rečima, na centralni server stižu vrednosti klasnih labela, dok se za potpuno centralizovane obrade mora voditi računa o tipovima senzora na različitim čvorovima, lokacijama čvorova, itd.



Slika 27: Struktura sistema na kojem se izvršava procedura distribuiranog mašinskog učenja. Umesto jedne centralizovane analize, podaci se prvo analiziraju posebno na svakom čvoru, pa se zatim rezultati lokalnih analiza spajaju u globalni rezultat.

Kako brojevi senzora rastu sa razvojem tehnologije (o čemu je bilo reči u uvodnom delu ove teze) klasifikacioni algoritmi postaju sve češći izbor kako bi se izašlo na kraj sa rezulutujućim količinama podataka. Distribuirane verzije predstavljaju zanimljivu alternativu centralizovanoj obradi, iz navedenih razloga.

3.1. Specifikacija problema

Najbolja rešenja za distribuirano mašinsko učenje u bežičnim senzorskim mrežama koriste jednostavne klasifikatore lokalizovane na čvorovima, čime se omogućava da svaki čvor donese lokalnu odluku na osnovu očitavanja sa senzora koji se nalaze na njemu. Te odluke se potom šalju do centralne lokacije

gde se na osnovu njih donosi globalna odluka. Količina komunikacije u mreži je na ovaj način smanjena, pošto se ne šalju celokupna očitavanja sa senzora, već samo odluke iz lokalnih klasifikatora, dok je obrada na centralnoj lokaciji pojednostavljena.

Ovakva obrada se oslanja na sposobnost pojedinačnih čvorova da donesu ispravne odluke samo na osnovu podataka koji su lokalno dostupni. U slučaju da postoje značajne interakcije između podataka koji se očitavaju sa senzora na različitim senzorskim čvorovima, dolazi do gubitka dela informacija. Pod interakcijama se misli na postojanje zakonitosti koje su vidljive samo ako se dva ili više podataka istovremeno posmatraju [Jakulin, 2004]. Ekstremni slučaj interakcije je XOR kolo [Jakulin, 2003], u kom slučaju nijedan čvor ne može da donese ispravnu odluku, osim ukoliko se do nje ne dođe slučajno.

U Tabeli 3 je prikazana binarna XOR funkcija. Neka su X_1 i X_2 veličine koje se mogu direktno izmeriti pomoću nekih senzora, a neka vrednost XOR funkcije označava da li se dogodio događaj koji se želi detektovati (na primer vrednost 0 znači da se događaj nije dogodio, a vrednost 1 znači da se događaj dogodio).

Ukoliko se senzori za vrednosti za X_1 i X_2 nalaze na istom čvoru, lako je napraviti model koji će izračunavati vrednost XOR funkcije. Međutim, pretpostavimo da se senzori za vrednosti za X_1 i X_2 nalaze na različitim čvorovima, i da se želi koristiti distribuirani klasifikator. Dakle, oba senzora će pokušati da pronađu klasifikator koji bi, bar delimično, odredio vrednost XOR funkcije na osnovu svoje lokalne promenljive X . Ako se pogleda Tabela 3, jasno je da nijedan od čvorova ne može da donese nikakav zaključak, osim da donese nasumičnu odluku. Ceo sistem će time raditi na nivou tačnosti nasumičnog pogađanja.

Tabela 3: XOR funkcija

X_1	X_2	$X_1 \text{ xor } X_2$
0	0	0
0	1	1
1	0	1
1	1	0

Postojeća rešenja takođe nisu pogodna za slučajeve kod kojih se posmatraju događaji koji su приметni ukoliko se posmatraju odluke sa nekog podskupa skupa čvorova u sistemu, koji je znatno manji od skupa svih čvorova. Kako senzorske mreže postaju veće i kompleksnije, oba ova problema će dobijati na važnosti. U tekstu koji sledi se razmatra pristup koje nudi rešenje za ove probleme.

3.2. Opis predloženog rešenja

U poglavlju koje sledi biće data preciznija specifikacija problema koji se posmatra, dok će uslovi i pretpostavke koji su usvojeni pri razvoju predloženog rešenja biti objašnjeni i obrazloženi. Potom će biti dat detaljan opis predloženog rešenja.

3.2.1. Scenario predviđene primene senzorske mreže za koju se razvija algoritam distribuiranog mašinskog učenja

Bežična senzorska mreža je namenjena nadgledanju neke oblasti sa ciljem otkrivanja izvesnog skupa događaja. Mreža je sinhrona, odnosno svi senzorski čvorovi periodično uzimaju merenja sa svojih senzora, u približno istim trenucima. Savršena fizička sinhronizacije nije neophodna, već je dovoljno da postoji sinhronizacija na nekom logičkom nivou. Drugim rečima, očitavanja sa senzora na različitim čvorovima se mogu upariti tako da opisuju strane u okruženju u jednom trenutku ili kratkom periodu vremena. Svaki čvor gradi lokalni klasifikator pomoću kojeg se može proceniti koji se od mogućih događaja najverovatnije desio neposredno pred očitavanja vrednosti sa senzora

(situacija u kojoj se nijedan događaj nije dogodio se takođe posmatra kao tip događaja). Dakle, čvorovi imaju različite klasifikatore, pri čemu svaki čvor sadrži klasifikator koji je napravljen prema njegovim lokalnim podacima. Izlazi iz lokalnih klasifikatora se propagiraju do centralnog servera, gde se pomoću njih formira globalna procena o eventualnom događaju u oblasti pokrivenom senzorskom mrežom.

3.2.2 Uslovi i pretpostavke

Svaki senzorski čvor sadrži tačno jedan senzor.

Ova pretpostavka je uvedena kao najnepovoljniji mogući scenario, s obzirom na to da svaki senzorski čvor mora da donese lokalnu odluku na osnovu vrednosti samo jednog atributa. Povećanje broja senzora po čvoru bi moglo da poveća tačnost rada sistema.

Samo jedan događaj može da se dogodi u toku jednog logičkog ciklusa.

Ova pretpostavka povlači da u svakom ciklusu postoji tačno jedan ispravan rezultat klasifikacije. Ukoliko bi se dozvolilo da ima više tačnih odgovora, problem bi se zakomplikovao, pošto bi umesto jednog tačnog odgovora trebalo odabrati skup odgovora. Alternativno, problem bi se mogao rešiti uvođenjem veštačkih složenih događaja, od kojih bi svaki predstavljao kombinaciju prostih realnih događaja. Ovo drugo rešenje bi zahtevalo domensko znanje pomoću kojega bi se odredilo koje kombinacije realnih događaja mogu zajedno da se dogode, ili bi doveo do eksponencijalnog rasta broja klasa mogućih događaja.

Događaji se dešavaju samo na unapred poznatim lokacijama.

Ova pretpostavka znači da je događaj jedinstveno definisan parom (tip događaja, lokacija događaja). Alternativno, problem bi se mogao posmatrati kao dve odvojene klasifikacije, jedna za tip događaja i druga za lokaciju. Ukoliko bi lokacije događaja mogle biti i kontinualne, one bi se umesto klasifikatorom mogle određivati regresionim modelom (što je teži problem za rešavanje).

Ukoliko lokacije događaja nisu od interesa, ova pretpostavka se može zanemariti.

Početni skup očitavanja sa svih senzora, zajedno sa podatkom o pridruženim događajima je raspoloživ na centralnom serveru.

Ovakav skup podataka se može dobiti na osnovu praćenja rada sistema u nekom reprezentativnom periodu, bez aktivnog algoritma mašinskog učenja, pri čemu se sva očitavanja šalju do centralnog servera i ručno označavaju. Alternativno, događaji se mogu veštački izazivati, ili se do podataka može doći simulacijama rada sistema.

3.2.3. Algoritam rada sistema

Algoritam rada sistema je sličan onome koji je opisan u radu pod referencom [McConnell, 2005]. U toku treniranja svakom čvoru se šalje početni skup podataka na osnovu kojeg će graditi lokalni klasifikator, uz pridružene vrednosti klasnih labela (informacije o eventualnim događajima u oblasti koja se nadgleda mrežom). Pošto su napravljeni svi lokalni klasifikatori, prelazi se u fazu upotrebe u kojoj se vrše procene da li se i koji događaj desio u oblasti koju pokriva senzorska mreža. Prave se lokalna predviđanja na osnovu lokalnih klasifikatora i podataka dobijenih sa senzora. Globalna odluka se na centralnom serveru donosi većinskom logikom, odnosno procedurom „glasanja“. Svaka odluka iz lokalnog klasifikatora se posmatra kao „glas“, pa se za globalnu odluku proglašava ona koja se najčešće pojavljuje među lokalnim odlukama. Glasanje može biti obično, pri čemu svaki čvor dobija jednaku vrednost glasa, ili težinsko, kada se vrednost glasa svakog čvora određuje prema poverenju koje imamo u ispravnost njegovog predviđanja.

3.2.4. Određivanje vrednosti glasova pojedinih čvorova

Vrednosti glasova pojedinih čvorova u procesu glasanja se mogu odrediti na više načina. Obično glasanje, gde je vrednost svakog glasa jednaka 1, i težinsko

glasanje na osnovu istorijske tačnosti klasifikatora poznatih od ranije [McConnell, 2004], dok se kao jedna od naučnih kontribucija ove teze uvode i težinsko glasanje na osnovu F1 metrike [VanRijsbergen1979] i težinsko glasanje na osnovu kapa-statistike [Berry1992]. Koeficijenti za sve pomenute šeme težinskog glasanja se izračunavaju na svakom čvoru posebno, upotrebom celog skupa podataka korišćenog za treniranje, posebnog skupa za testiranje performansi ili nekom vrstom kros-validacije (na primer desetostrukom kros-validacijom, ili izostavljanjem po jednog primera).

a) Obično glasanje

Kod običnog glasanja svaki senzorki čvor dobija glas jednake vrednosti. Za globalnu odluku se proglašava ona koja se najčešće javlja među lokalnim odlukama u trenutnoj iteraciji glasanja.

b) Težinsko glasanje na osnovu tačnosti lokalnih klasifikatora

Kod ovakvog glasanja se kao vrednost glasa svakog čvora uzima procenat koliko puta je dati čvor ispravno dao predviđanje koje trenutno daje, gledano u odnosu na sve slučajeve u kojima je dao to predviđanje. Ovakav izbor težinskih faktora se bazira na relativno jednostavnoj i intuitivnoj metrici, i , kao što će biti pokazano, daje bolje rezultate od običnog glasanja.

c) Težinsko glasanje na osnovu F1 metrike

F1 metrika je mera performansi klasifikatora koja predstavlja harmonijsku sredinu osetljivosti i preciznosti klasifikatora:

$$F1 = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}}$$

Preciznost predstavlja broj instanci podataka koje pripadaju datoj klasi i koje je klasifikator je označio kao pripadnike datoj klasi podeljen sa ukupnim brojem instanci koje je klasifikator označio kao da pripadaju datoj klasi. Osetljivost je broj instanci koje pripadaju datoj klasi i koje je klasifikator označio kao

pripadnike date klase, podeljen ukupnim brojem instanci koje pripadaju datoj klasi.

F1 se smatra robusnijom merom performansi klasifikatora od tačnosti, pre svega zato što je otpornija na slučajeve kod kojih tačnost može biti visoka, iako klasifikator dosta greši. Primena metrike kao težinskog faktora pri glasanju velikog broja klasifikatora, kao što je ovde slučaj, iziskuje da se izračunavanja i poređenja često vrše automatski, bez ručne analize dobijenih brojki od strane stručnjaka. Samim tim, nameće se potreba za korišćenjem što robusnijih metrika, kako bi se pomenuti specifični slučajevi što ređe dešavali, pa je logično uzeti i F1 metriku u razmatranje.

3.2.5. Težinsko glasanje na osnovu kapa-statistike

Kapa-statistika je mera performansi klasifikatora koja uzima u obzir da će u određenom broju slučajeva klasifikator sasvim slučajno doneti ispravnu odluku.

Kapa-statistika se računa kao:

$$k = \frac{P(a) - P(e)}{1 - P(e)}$$

gde je $P(a)$ tačnost klasifikatora (procenat ispravnih odluka o vrednosti labele), a $P(e)$ tačnost klasifikatora koja se može očekivati ukoliko bi se odluke donosile potpuno nasumično. Vrednosti kapa-statistike se kreću u rasponu $[-1, 1]$, ali pošto se u šemama glasanja primenjuju kao težinski faktori sve vrednosti manje od 0 se dodeljuje vrednost 0. Na ovaj način se sprečava da neki čvor glasa sa negativnom težinom, što ne bi imalo smisla.

3.2.6. Koraci pri izvršavanju algoritma

Faza treniranja sistema, odnosno lokalnih klasifikatora se može prikazati u nekoliko koraka, kao što je dato u pseudokodu na Slici 28.

Algoritam 1

For each node do

The server site sends the measurements corresponding to the node's sensors
and the target value to the node

The node builds a local predictor based on this data

If Voting scheme other than Simple voting is used

The node computes the appropriate weighting measure
and reports the values to the central site

End If

End For

Slika 28. Pseudokod koji prikazuje treniranje klasifikatora u okviru sistema

Po završenom treniranju sistem može da uđe u upotrebnu fazu, kada se koristi za nadgledanje oblasti koju pokriva i otkrivanje događaja koji su definisani kao labela za klasifikaciju. Rad sistema se može prikazati u nekoliko koraka, što je dato na Slici 29, u formi pseudokoda.

Algoritam 2

For each node do

The node takes the readings from its local sensors.

The node's local predictor is used with the readings to make a local prediction.

The node sends the local prediction to the central server.

If Voting scheme other than Simple voting is used

Find the appropriate weight associated with the node
and its local prediction.

End If

End For

Slika 29. Pseudokod koji prikazuje rad sistema

3.2.7. Pravljenje lokalnih modela koji uključuju podatke sa većeg broja čvorova kada među podacima postoje značajne interakcije

U slučajevima u kojima postoje značajne interakcije među podacima koji se dobijaju sa senzora koji su na različitim čvorovima, moguće je dobiti znatno bolje predikcije ukoliko se, umesto potpuno lokalnih, prave jedinstveni klasifikatori na osnovu ovakvih podataka, spojenih u jedan skup podataka. Drugim rečima, mogu se dobiti bolji lokalni klasifikatori ukoliko se podaci sa nekoliko čvorova skupe na jednom čvoru, pa onda gradi klasifikator na osnovu svih tih podataka. Sa druge strane, ovakva procedura zahteva slanje i prijem senzorskih podataka, čime se troši dodatna energija i odustaje od očuvanja privatnosti na nivou čvora. U nekim situacijama ovakav postupak je ipak ili neophodan ili veoma koristan. Kao još jedan naučni doprinos ove teze, razvijen je algoritam koji omogućava treniranje lokalnih klasifikatora na osnovu podataka sa više čvorova. Kako bi se izbegla nepotrebna komunikacija, vodilo se računa da se podaci kombinuju samo tamo gde interakcije zaista postoje. Takođe, uzima se u obzir samo mogućnost interakcija između podataka sa neposredno susednih čvorova.

U toku treniranja se početni skupovi podataka šalju do odgovarajućih čvorova, isto kao u osnovnom algoritmu, bez spajanja podataka među kojima postoji interakcija. Čvorovi zatim ove podatke razmenjuju sa sebi susednim čvorovima. Svaki čvor će napraviti i potpuno lokalni klasifikator (kao u osnovnom algoritmu) i klasifikator koji uzima u obzir i podatke sa susednih čvorova. Uz to, svaki čvor gradi i kopije lokalnih klasifikatora sa okolnih čvorova (onako kako bi svaki od okolnih čvorova izgradio klasifikator, imajući samo svoje podatke). Rezultati potpuno lokalnih i grupnog klasifikatora se upoređuju, tako što se potpuno lokalni klasifikatori spajaju globalnom procedurom glasanja. Ukoliko je grupni klasifikator značajno bolji, on se usvaja. Postupak glasanja i izgradnja lokalnih klasifikatora na osnovu susedskih podataka se izvodi kako bi se uočila razlika koja će rezultovati u globalnom modelu.

Faza treniranja sistema, odnosno lokalnih klasifikatora se može prikazati u nekoliko koraka, kao što je dato u pseudokodu na Slici 30.

Algoritam 3

```
For each node do
    Send the measurements corresponding to the node's sensors
        and the target value from the server site.
    The node builds a local predictor based on the local data.
    The node sends its data to its immediate neighbors.
    The node waits to receive data from all of its immediate neighbors.
    The node builds a group predictor, based on both the local data
        and the data from its neighbors.
If the group predictor performs better than the local predictor
        The node selects the group predictor and notifies its neighbors
End If
If a voting scheme other than Simple voting is used
        The node computes the appropriate weighting measure
        and reports the values to the central site
End If
End For
```

Slika 30. Pseudokod koji prikazuje treniranje klasifikatora u okviru sistema

Po završenom treniranju sistem može da uđe u upotrebnu fazu, kada se koristi za nadgledanje oblasti koju pokriva. Rad sistema takođe sadrži neke izmene u odnosu na osnovnu verziju algoritma, što se može videti na Slici 31, u formi pseudokoda.

Algoritam 4

```
For each node do
    The node takes the readings from its local sensors.
If a neighboring node sent a notification that it has a group predictor
    using this node's sensor readings.
        Node sends the readings to the appropriate neighbor.
Else
If the node runs a group predictor with values from its neighbors
        Wait for the readings from the neighboring nodes.
End if
    The node uses the available readings to make a local (or group) prediction.
    The node sends the prediction to the central server.
If Voting scheme other than Simple voting is used
        Find the appropriate weight associated with the node and its
prediction.
```

End If
End If
End for

Slika 31. Pseudokod koji prikazuje rad sistema kada se koriste grupni klasifikatori na lokalnom nivou

3.3. Analiza rada sistema na osnovu simulacije sa skupovima podataka

U nastavku je data analiza rada sistema u kojem bi bio implementiran opisani algoritam.

3.3.1. Metodologija

Simuliran je rad opisanog sistema u bežičnim senzorskim mrežama čije su topologije nasumično generisane. Upotrebljena je RANDOM procedura generisanja topologija, opisana u sklopu GENSEN generatora mrežnih topologija [Camilo, 2007]. Atributi iz skupova podataka su nasumično dodeljivani pojedinim čvorovima, kao veličine koje se mogu direktno očitati sa njihovih senzora. Rad sistema je analiziran primenom desetostruke kros-validacije. Senzorski čvorovi lokalno takođe koriste desetostruku kros-validaciju kako bi izračunali eventualne metrike potrebne pri određivanju težinskih koeficijenata kod glasanja. Rezultati koji će biti predstavljeni su srednje vrednosti iz 10.000 ponovljenih simulacija. U svakoj simulaciji je generisana nova topologija mreže, dok su podaci ponovo raspoređeni po čvorovima.

3.3.2. Podaci koji su korišćeni prilikom simulacija

Simulacije su rađene sa pet skupova podataka iz UCI repozitorijuma skupova podataka za mašinsko učenje [Frank, 2010]. U ovim skupovima podataka broj atributa u skupu se kreće između 10 i 35, dok se broj instanci podataka kreće od 194 do 8.124. Korišćeni su skupovi podataka koji sadrže samo numeričke attribute, samo nominalne attribute, kao i kombinaciju ovih tipova atributa. U nastavku su dati opisi upotrebljenih skupova podataka.

Credit Approval

„Credit Approval“ skup podataka sadrži podatke o odobrenim i odbijenim zahtevima za kreditnim karticama. Skup sadrži šarenoliku kombinaciju različitih tipova atributa, uključujući realne i cele brojeve, kao i kategorijske promenljive sa malim i velikim brojem mogućih vrednosti.

Tabela 4: Detalji Credit Approval skupa podataka

Broj podataka	Broj atributa	Tipovi atributa	Nepoznate vrednosti
690	15	Nominalni, celobrojni, realni	Da

Glass Identification

Skup „Glass Identification“ sadrži podatke o sadržaju oksida u šest različitih tipova stakla. Cilj je odrediti o kojem tipu stakla se radi. Skup sadrži samo numeričke, realne podatke.

Tabela 5: Detalji Glass Identification skupa podataka

Broj podataka	Broj atributa	Tipovi atributa	Nepoznate vrednosti
214	10	Realni	Ne

Flags

„Flags“ skup podataka sadrži informacije o izgledu zastava i detalje vezane za zemlje kojima pripadaju. Problem koji je analiziran je predviđanje većinske religije zemlje na osnovu izgleda zastave.

Tabela 6: Detalji Flags skupa podataka

Broj podataka	Broj atributa	Tipovi atributa	Nepoznate vrednosti
194	30	Nominalni, celobrojni	Ne

Soyabean Large

„*Soyabean Large*“ je skup podataka koji se veoma često koristi pri evaluaciji algoritama mašinskog učenja. Cilj analize je klasifikacija zrna soja u četiri moguće kategorije bolesti, prema karakteristikama zrna. Svi atributi su nominalni.

Tabela 7: Detalji Soyabean Large skupa podataka

Broj podataka	Broj atributa	Tipovi atributa	Nepoznate vrednosti
307	35	Nominalni	Da

Mushrooms

„*Mushrooms*“ skup podataka sadrži informacije o karakteristikama 22 vrste pečuraka. Cilj analize je pogoditi koje pečurke su otrovne, a koje jestive. Sve vrednosti u skupu su nominalnog tipa.

Tabela 7: Detalji Mushrooms skupa podataka

Broj podataka	Broj atributa	Tipovi atributa	Nepoznate vrednosti
8124	22	Nominalni	Da

3.2.3. Rezultati simulacija i njihova analiza

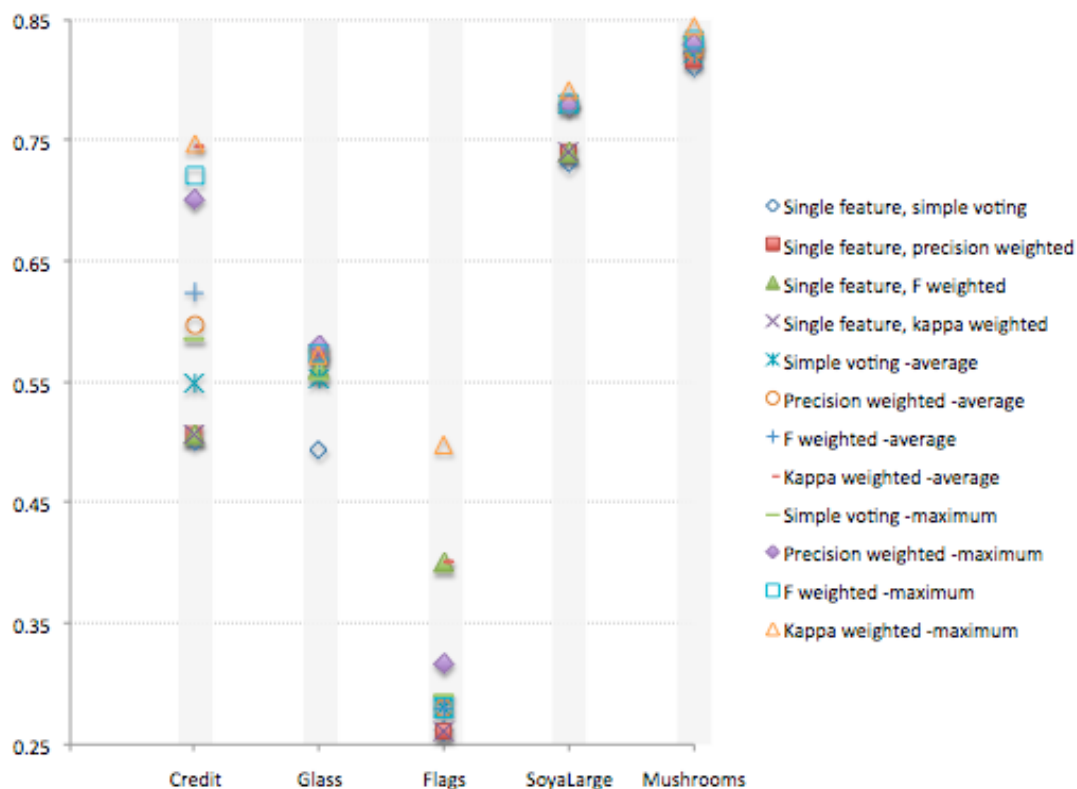
Na Slici 32 su dati rezultati klasifikacije u vidu prosečnih postignutih tačnosti, pri upotrebi različitih načina određivanja vrednosti glasova, sa kombinovanjem promenljivih kod kojih postoje interakcije i bez njih. Za šeme koje koriste kombinacije promenljivih kod kojih postoje interakcije su dati i najbolji rezultati. Razlog je što se u realnoj senzorskoj mreži može očekivati da promenljive sa visokim nivoom interakcija budu vezane za senzore na bliskim čvorovima, pa bi se time nalazili u podskupovima koji se mogu kombinovati.

Algoritmi koji koriste kombinacije promenljivih kod kojih postoje interakcije generalno rade bolje od šema gde se lokalni klasifikatori grade od podataka dobijenih samo sa lokalnih senzora. Proverom je utvrđeno da su najbolji

rezultati nastali u sistemima sa kombinovanim promenljivima, pri čemu su podaci kod kojih postoje jake interakcije dodeljeni susednim čvorovima.

Težinska glasanja pokazuju bolje rezultate od običnog glasanja. Glasanje sa težinskim koeficijentima izračunatim na osnovu F1-metrike daje bolje rezultate od glasanja sa težinskim koeficijentima izračunatim na osnovu tačnosti (procenta tačnih odogovra). Primena kapa-statistike pokazuje najbolje rezultate. Obrazloženje se može pronaći u tome što kapa-statistika, uz to što je robusna metrika performansi klasifikatora, može veoma efikasno uklanjati ili umanjivati glasove koji potiču od čvorova koji nisu značajno korelisani sa vrednosti klase podatka.

Za skupove podataka *Credit Approval*, *Flags* i *Soyabean Large* sistem radi znatno bolje kada se koristi verzija algoritma koja otkriva i koristi podatke sa bliskih čvorova kod kojih postoji visok nivo interakcija. Ovakav rezultat je u skladu sa očekivanjima, s obzirom na to da ovi skupovi podataka zaista sadrže visok nivo interakcija među atributima. Sa druge strane, kod *Mushrooms* i *Glass Identification* skupova podataka ove verzije algoritma daju slične rezultate. Ovakav rezultat je takođe u skladu sa očekivanjima, pošto ovi skupovi podataka ne sadrže visok nivo interakcija među atributima. U svim slučajevima najbolji rezultati se postižu ako se vrednosti glasova lokalnih klasifikatora određuju na osnovu vrednosti lokalne kapa-statistike.



Slika 32: Rezultati klasifikacije nad različitim skupovima podataka. Legenda: 'Single feature' označava sisteme koji koriste samo lokalne promenljive pri kreiranju klasifikatora na čvorovima, dok rezultati koji nisu označeni kao 'Single feature' uključuju i lokalne klasifikatore sa podacima sa više okolnih čvorova.

3.3. Analiza komunikacije u mreži

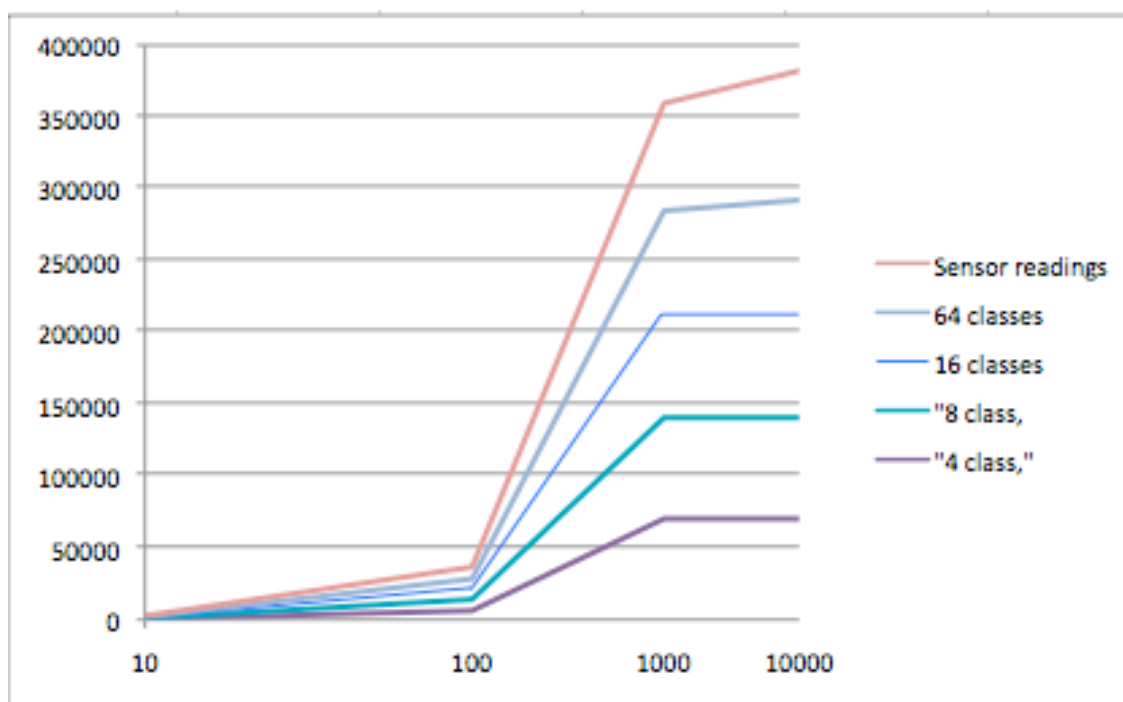
3.3.1. Metodologija

Analiza saobraćaja u bežičnoj senzorskoj mreži kada se koristi opisani algoritam distribuiranog mašinskog učenja, i radi poređenja, kada sistem radi na klasičan način (slanjem svih vrednosti očitanih sa senzora do centralnog servera) urađene su uz pomoć simulacija. Simuliran je jedan logički ciklus koji uključuje svu potrebnu komunikaciju kako bi se neki događaj otkrio na centralnom serveru. Ukoliko se koristi distribuirano mašinsko učenje, ovo uključuje slanje svih lokalnih predviđanja do centralnog servera, dok se u slučaju klasičnog rada šalju celokupni podaci očitani sa senzora. Pretpostavlja se da poruke koje se

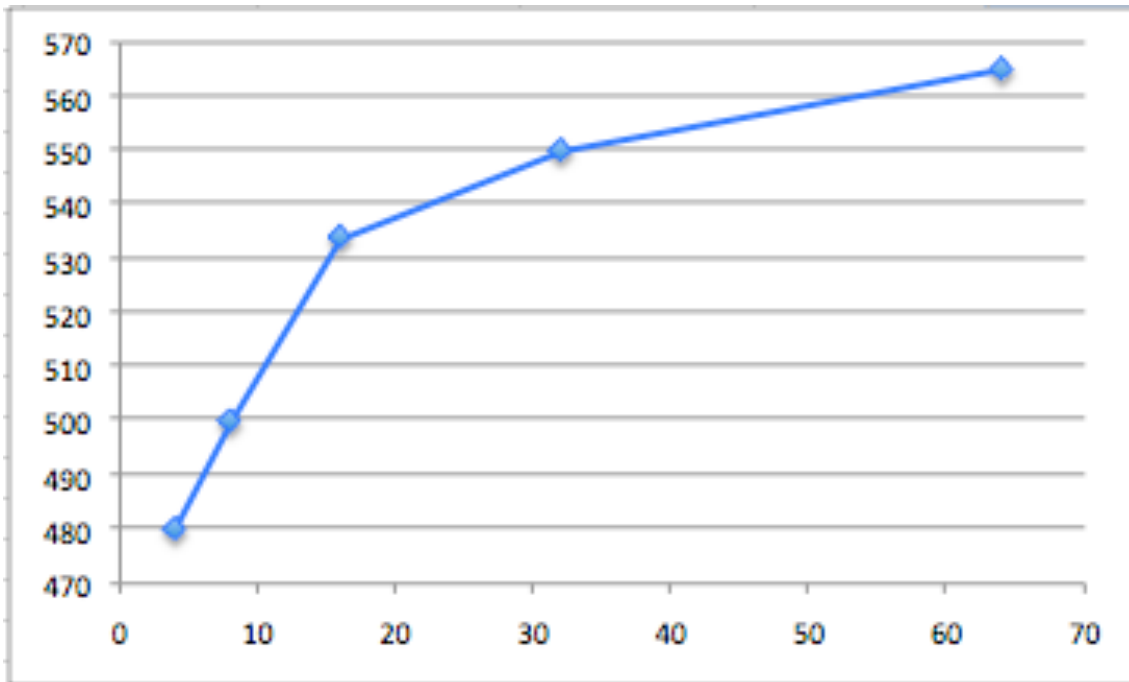
šalju sadrže adresu odredišta, adresu čvora sa kojega su potekle i sam podatak (očitanje sa senzora ili klasnu labelu koja je izlaz iz lokalnog klasifikatora). Pretpostavljeno je da su sve adrese veličine 16 bita, kao i da su klasne labele kodirane minimalnim neophodnim brojem bita ($\log_2(\text{broja mogućih vrednosti})$).

3.3.2. Rezultati analize komunikacije

Slika 33 prikazuje količinu saobraćaja u mreži (u vidu broja bitova) u sistemu koji radi na klasičan način prenošenjem očitavanja sa senzora i u različitim uslovima sistema koji rade na principu opisanog algoritma mašinskog učenja. Simulacije su rađene za mreže čije su veličine između 10 i 10.000 čvorova. Na Slici 34 je prikazana zavisnost količine komunikacije u mreži u toku jednog logičkog ciklusa u funkciji od broja vrednosti koje izlazi iz lokalnih klasifikatora mogu da uzmu (broja različitih tipova događaja koji se mogu desiti i koji se žele detektovati). Dati grafikon je nacrtan na osnovu primera mreže sa 10 čvorova.



Slika 33: Količina komunikacije u bežičnoj senzorskoj mreži (broj bitova koji se šalju i primaju) u toku jednog logičkog ciklusa, kao funkcija od broja čvorova u mreži



Slika 34: Količina komunikacije (broj bitova koji se šalju i primaju) u bežičnoj senzorskoj mreži sa deset čvorova, u toku jednog logičkog ciklusa, kao funkcija od broja različitih tipova događaja koji se mogu desiti i koji se žele detektovati

3.4. Potrošnja energije u toku rada sistema

Kao što je već više puta ponovljeno, veliki deo energije koji se potroši u toku rada jedne bežične senzorske mreže odlazi na potrošnju pri komunikaciji između čvorova. Sa druge strane, u predloženom rešenju se javlja dodatna potrošnja u vidu procesiranja podataka na čvorovima, odnosno treniranja i upotrebe lokalnih modela mašinskog učenja. U tekstu koji sledi biće data analiza potrošnje energije koja rezultuje u sistemu i odnosa koji se uspostavljaju između ostvarenih ušteda u komunikacija i dodatne potrošnje pri procesiranju.

Kako bi izveli analizu, uvešćemo nekoliko parametara. Neka je sa N označena veličina skupa podataka koji se koristi pri treniranju lokalnih klasifikatora (broj instanci podataka u ovom skupu). Neka je C broj različitih tipova događaja koji se mogu primetiti u oblasti koja se nadgleda mrežom (odnosno broj različitih vrednosti koje se mogu naći na izlazu iz lokalnih klasifikatora). Neka je M broj različitih senzora na jednom čvoru (pretpostavićemo da su svi čvorovi isti i

svaki sadrži M različitih senzora), a neka je D prosečno rastojanje između čvorova i mesta gde podaci napuštaju mrežu, prema topologiji mreže (drugim rečima, D je prosečan broj čvorova koji moraju da posreduju u komunikaciji kada se podaci šalju sa nekog čvora do centralnog servera). Sa E_{cp} ćemo predstaviti energiju koja je potrebna da bi se na senzorskom čvoru izvršila jedna instrukcija, dok je E_{cm} energija potrebna da se jedna jedinica podataka prenese sa jednog senzorskog čvora na drugi, u direktnoj komunikaciji.

Računska složenost algoritma J48, koji se koristi kao lokalni klasifikator u ovom radu je data sa (prema referenci [Lim2000]):

$$E_{train} \cong M \cdot N \cdot \log_2(N) \cdot E_{cp}$$

dok je ukupna energija koja je potrebna da bi se izvršila jedna lokalna klasifikacija i rezultat poslao do centralnog servera data kao:

$$E_{op} \cong \log_2(M) \cdot E_{cp} + D \cdot \log_2(C) \cdot E_{cm} \approx D \cdot \log_2(C) \cdot E_{cm}$$

Sa druge strane, neka je R broj jedinica podataka sa jednog senzora koje se prenose kroz mrežu ukoliko se koristi klasičan pristup slanja celokupnih očitavanja sa senzora. Prema tome, za M senzora koji se nalaze na jednom čvoru potrebno je utrošiti energiju:

$$E_{trad} \cong D \cdot R \cdot M \cdot E_{cm}$$

Da bi opisani algoritam obezbedio manju potrošnju energije pri radu u poređenju sa klasičnim sistemom neophodno je da bude ispunjen uslov, posle Z logičkih ciklusa rada:

$$E_{train} + Z \cdot E_{op} < Z \cdot E_{trad}$$

Posle relativno jednostavnog izvođenja iz jednačina 1□4 dobijamo uslov koji daje minimalni broj logičkih iteracija rada posle kojih opisani algoritam postaje energetska efikasniji od klasičnog rešenja (drugim rečima, broj iteracija rada sistema posle kojih je energija utrošena pri treniranju kompenzovana uštedama pri komunikaciji):

$$Z > \frac{M \cdot N \cdot \log_2(N) \cdot \frac{E_{cp}}{E_{cm}}}{D \cdot (R \cdot M - \log_2 C)}$$

Kako bi stvorili jasniju sliku o ovome rezultatu, uzećemo jedan primer sa konkretnim brojkama, na osnovu realnih uređaja koji su široko dostupni. U Tabeli 8 je data energetska cena slanja i prijema podataka, kao i potrošnja energija pri izvršavanju instrukcija za senzorski čvor TelosB [DeMeulenaer2008].

Tabela 8: Utrošak energije pri radu TelosB senzorskog čvora

Operacija	Potrebna energija
Izvršavanje jedne instrukcije (prosek)	6 nJ
Slanje jednog bita podataka	0.72 μJ
Prijem jednog bita podataka	0.81 μJ
Prenos jednog bita podataka	1.52 μJ

Neka se mreža sastoji od senzorskih čvorova koji sadrže po četiri različita senzora povezana preko 12-bitnog AD konvertora. Neka lokalni skupovi podataka na osnovu kojih se kreiraju lokalni klasifikatori sadrže po 100 instanci podataka, i neka postoji 16 mogućih tipova događaja u sredini koja se nadgleda bežičnom senzorskim mrežom. Pretpostavimo da su svi čvorovi u blizini tačke preko koje podaci napuštaju mrežu. U ovom slučaju je potrošnja energije pri komunikaciji minimalna, dok je potrošnja energije pri izračunavanjima potrebnima za treniranje lokalnih klasifikatora ista, pa je ovakav slučaj najpovoljniji za klasičan način rada mreže u poređenju sa opisanim pristupom sa mašinskim učenjem.

Prostim unosom brojeva u nejednačinu (5) dobijamo da je nejednakost zadovoljena ako je $Z < 7$. Ovakav rezultat praktično znači da se energija koja je potrošena kako bi se izračunali lokalni klasifikatori nadoknadi već prilikom slanja podataka u prvom logičkom ciklusu, pošto je količina podataka koja se šalje i prima znatno manja (4 umesto 48 bita po jednom senzorskom čvoru).

Možemo uzeti u razmatranje i uticaj pojedinačnih parametara na potrošnju energije u sistemu. Ukoliko se vratimo na nejednačinu (5) i napišemo je u razvijenom obliku, dobijamo:

$$M \cdot N \cdot \log_2(N) \cdot E_{cp} + Z \cdot D \cdot \log_2(C) \cdot E_{cm} < Z \cdot D \cdot R \cdot M \cdot E_{cm}$$

Leva strana nejednačine koja predstavlja opisano rešenje sa mašinskim učenjem se sastoji od dva elementa. Prvi element je vezan za energetska cenu treniranja lokalnih algoritama, dok je drugi vezan za cenu prenosa podataka do centralnog servera. Energija koja se utroši prilikom treniranja lokalnih klasifikatora zavisi od veličine skupa podataka koji se koristi za treniranje i broja senzora na čvoru. Međutim, u realističnim uslovima (senzorski čvorovi sa desetak različitih senzora, nekoliko stotina senzora u svakom lokalnom skupu podataka, brojevi iz Tabele 8) odnos E_{cp}/E_{cm} je veoma mali, pa time drugi element leve strane, koji je vezan za komunikaciju, dominira. Prema tome, ponašanje sistema će, u smislu potrošnje energije, pratiti količinu komunikacije, što je analizirano u prethodnom poglavlju.

3.5. Analiza rada sistema kroz simulaciju realnog okruženja

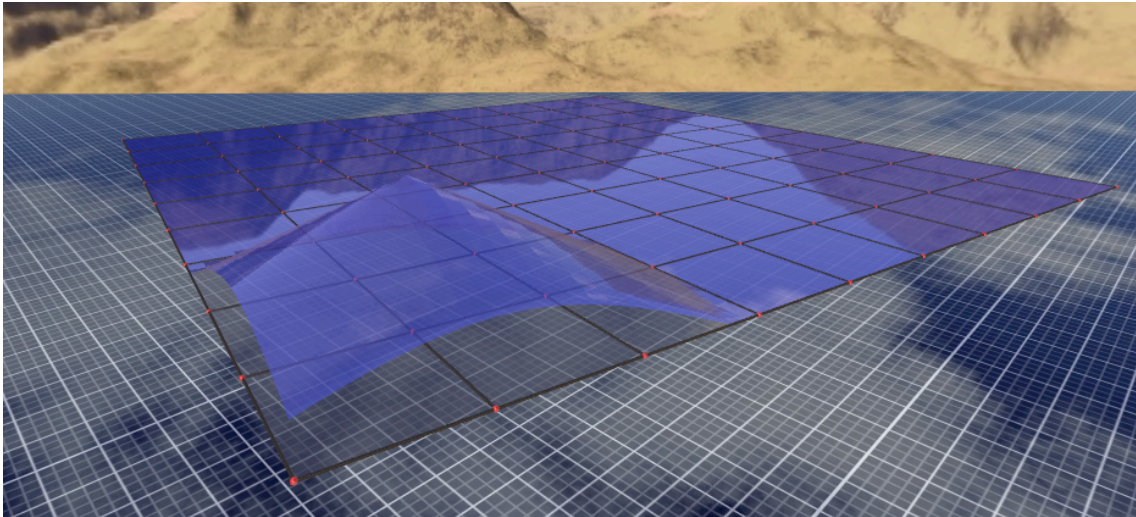
Uz analizu rada sistema na osnovu simulacije sa javno dostupnim skupovima podataka, što je opisano u prethodnom poglavlju, sistem je analiziran i kroz simulaciju rada u realnom okruženju. Pod realnim okruženjem se smatra simulacija bežične senzorske mreže čija namena odgovara očekivanoj primeni opisanog algoritma: detekcija diskretnih događaja u oblasti koja se nadgleda.

3.5.1. Metodologija

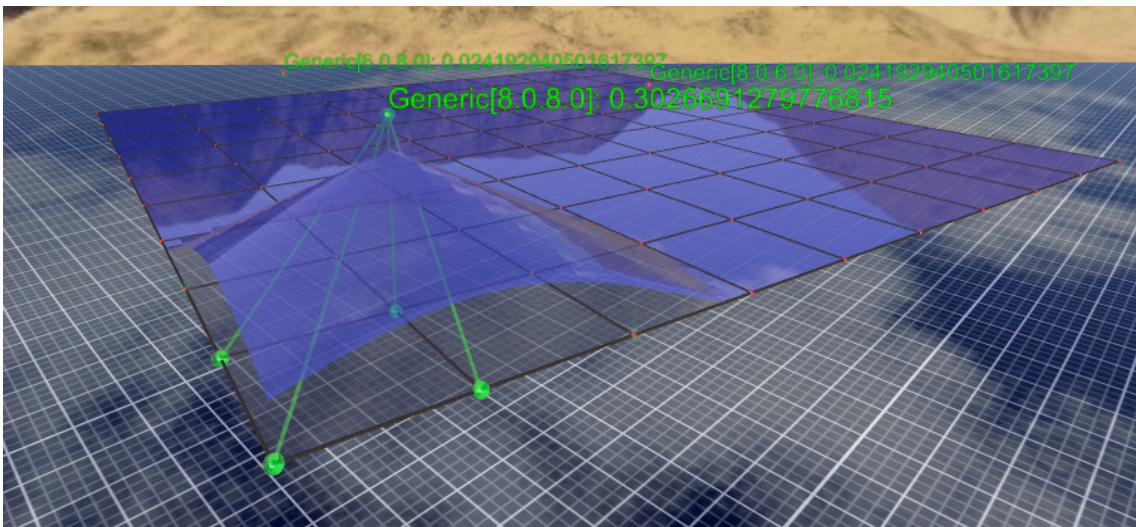
Simulirana je bežična senzorska mreža koja je postavljena tako da formira pravilnu, potpunu meš topologiju, dimenzija 10x10, odnosno ukupno 100 senzorskih čvorova. Pretpostavlja se da je svaki čvor opremljen jednim senzorom kojim može direktno da meri efekte koje proizvode događaji u okruženju. „Događaj” je simuliran kao promena u očitavanjima senzora, na nekoj lokaciji u polju koje pokriva mreža. Pretpostavljeno je da efekti događaja opadaju linearno sa odstojanjem od lokacije na kojoj se događaj desio. Simulacije su vršene u uslovima sistema u okruženju u kojem nema šuma, kao i u šumovitom okruženju. Simulirano je samo glasanje na osnovu kapa-statistike.

Simulacije u okruženju bez šuma

Pri simulacijama u sredini bez šuma jedina varijacija između iteracija simulacije je u lokaciji na kojoj se događaji dešavaju. Na Slici 35 je prikazano okruženje u kojem se nalazi bežična senzorska mreža, u trenutku u kojem se došlo do događaja u tački sa koordinatama (8,8). Slika 36 prikazuje različite odluke koje su donesene na različitim senzorskim čvorovima. Tri lokacije su dobile ne-nulte vrednosti glasova, i to su tačke (8,8), (8, 6) i (6,8). Ispravna lokacija (8,8) je dobila najviši rezultat.



Slika 35: Okruženje bez prisustva šuma u kojem je postavljena bežična senzorska mreža u trenutku kada je generisan događaj na lokaciji (8,8).



Slika 36: Odluke donete u sistemu po generisanom događaju na lokaciji (8,8) u okruženju bez prisustva šuma.

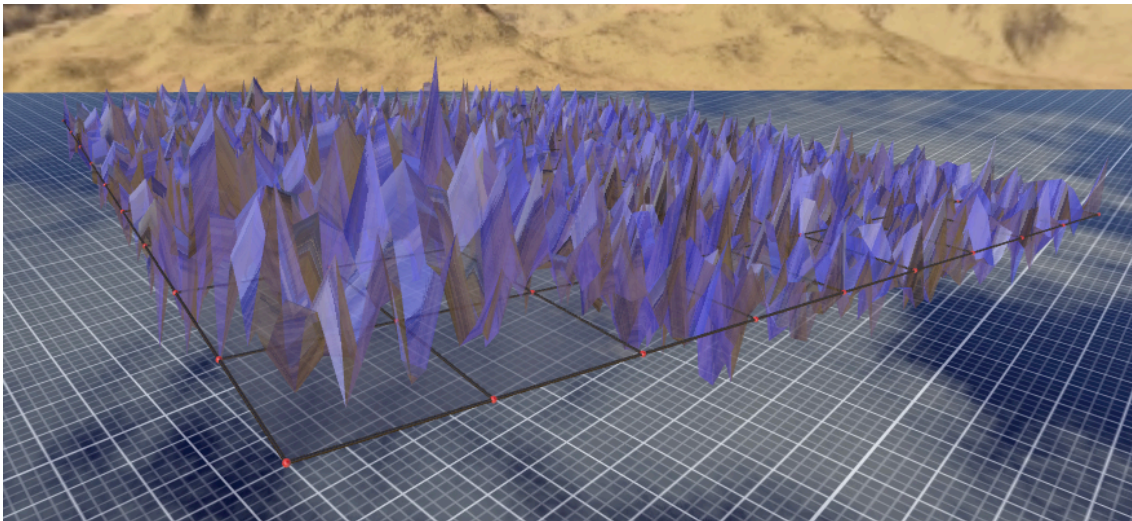
U okruženju u kojem nema šuma rad sistema zavisi samo od rezolucije bežične senzorske mreže (Tabela 9). Rezoluciju mreže definišemo kao rastojanje između lokacija na kojima mogu da se dese događaji, podeljeno sa rastojanjem između senzorskih čvorova. Uz primenu opisanog algoritma mašinskog učenja solidni rezultati se ostvaruju za vrednosti rezolucije 0.5 i veće. Za manje rezolucije tačnost je smanjena, ali su odluke i dalje u okolini pravih lokacija na kojima se događaji dešavaju.

Tabela 9: Rezultati rada algoritma vertikalno distribuirane klasifikacije u okruženju bez šuma

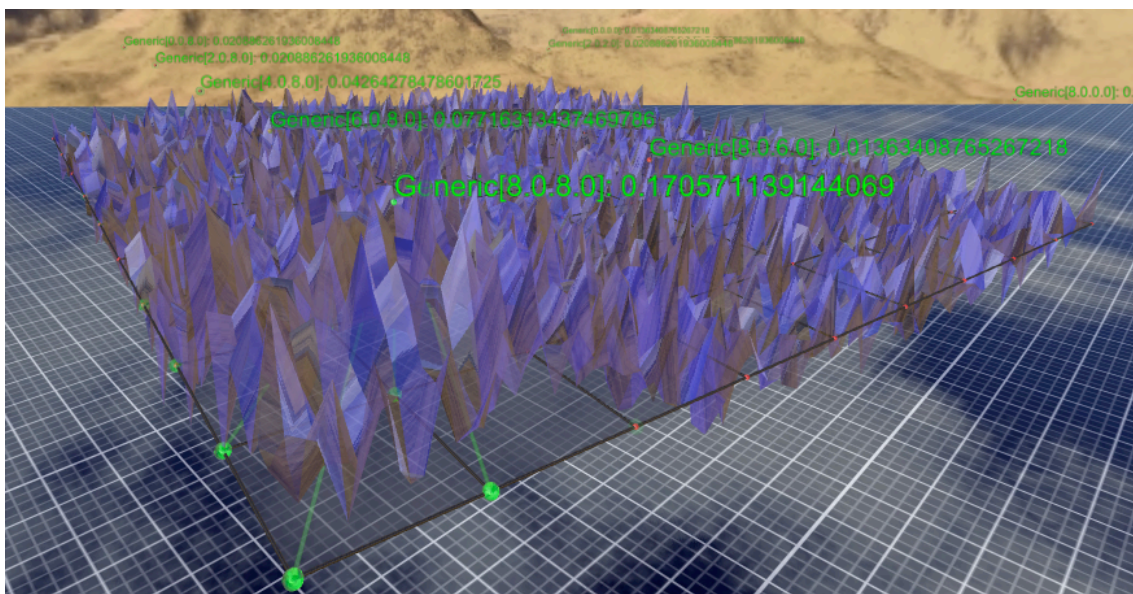
Rezolucija mreže	Tačnost sistema
2	0.97
1	0.97
0.5	0.83
0.25	0.44

Simulacije u okruženju sa prisustvom šuma

Sve simulacije rada opisanog sistema u okruženju sa prisustvom šuma su rađene uz rezoluciju senzorske mreže od 1. Korišćen je beli aditivni šum, tako što je za svaki senzor nasumično generisana vrednost iz normalne raspodele i dodavana vrednost koja se očitava sa senzora na datom čvoru. Na Slici 37 je prikazano okruženje sa prisustvom šuma u kojem se desio događaj koji se želi detektovati. Slika 38 prikazuje različite odluke koje su donesene na različitim senzorskim čvorovima. U poređenju sa sistemom koji radi u okruženju bez šuma primetan je znatno veći broj lokacija koje su dobile ne-nulte rezultate. Lokacija na kojoj se zaista odigrao događaj je još uvek ispravno označena kao rezultat sa najvećom sumom glasova, ali sa znatno manjom marginom.



Slika 37: Okruženje sa prisustvom šuma u kojem je postavljena bežična senzorska mreža u trenutku kada je generisan događaj na lokaciji (8,8).



Slika 39: Odluke donete u sistemu po generisanom događaju na lokaciji (8,8) u okruženju sa prisustvom šuma.

Za relativno male odnose signal-šum sistem pokazuje relativno dobre rezultate (Tabela 10), u smislu visoke tačnosti. Međutim, ukoliko je u sistemu odnos signal-šum velik, neophodna je dodatna obrada signala i podataka, ili unapređenja u algoritmu mašinskog učenja. Jedna mogućnost za ovakvo unapređenje je u postavljanju minimalnog praga vrednosti glasa koji se uzima u obzir. U opisanom sistemu glasovima se dodeljuje vrednost na osnovu kapastatistike, s tim da se vrednosti manje od 0 zaokružuju na nultu vrednost. U šumovitim okruženjima se ovo može proširiti, tako da se na primer vrednosti manje od 0.1 zaokružuju na 0. Na ovaj način se uzimaju u obzir samo glasovi čvorova koji su „sigurniji” u svoje odluke, te je manja verovatnoća da su one poremećene šumom. Treba uzeti u obzir da je pojam dobrih ili loših performansi u velikoj meri vezan za samu aplikaciju.

Tabela 10: Rezultati rada algoritma vertikalno distribuirane klasifikacije u okruženju sa prisustvom šuma

Odnos signal/šum	Tačnost sistema
0.05	0.93
0.1	0.84
0.15	0.77
0.25	0.52

3.6. Diskusija predloženog rešenja za vertikalno distribuiranu klasifikaciju u bežičnim senzorskim mrežama

Osnove opisanog algoritma potiču iz oblasti distribuiranog mašinskog učenja, pri čemu je većina metoda razvijana sa ciljem primene sa distribuiranim bazama podataka, a ne bežičnim senzorskim mrežama. Pažljiva analiza detalja vezanih za datu tematiku, u kombinaciji sa specifičnostima bežičnih senzorskih mreža, otkrila je da neke od osnovnih pretpostavki treba ponovo razmotriti i izmeniti.

Najvažnije primene opisanog algoritma se mogu očekivati u detekciji i karakterizaciji različitih događaja u oblasti koja se nadgleda uz pomoć bežične senzorske mreže. U velikim, geografski široko rasprostranjenim bežičnim senzorskim mrežama se može očekivati da će pojedini događaji biti otkriveni samo na malom podskupu čvorova, onima koji su fizički u blizini lokacije događaja. Ostali čvorovi (kojih je većina) će odluke donositi na osnovu očitavanja sa senzora koji nisu detektovali promene usled događaja (zbog udaljenosti od lokacije događaja). Prema tome, jednostavno glasanje nije pogodno za primenu sa bežičnim senzorskim mrežama, osim u slučajevima u kojima se može garantovati da će svi (ili bar većina) čvorovi biti u stanju da detektuju sve događaje u oblasti koju mreža pokriva. Može se očekivati da tačnost i F1 metrika daju nešto bolje rezultate, ali ukoliko čvorovi koji ne vide događaj čine veliku većinu, sistem takođe neće ispravno raditi. Kapa-statistika sa druge strane je u stanju da eliminiše glasove koji nastaju usled nasumičnog pogađanja, odnosno kada lokalni klasifikatori rade sa podacima koji nisu korelisani sa stvarnim događajem. Dakle, u sistemima bežičnih senzorskih mreža koji pokrivaju velike oblasti, i gde relativno mali broj čvorova može da detektuje događaje, treba koristiti glasanje na osnovu kapa-statistike. U Tabeli 9 data je ilustracija težinskih faktora koji se dobijaju sa nekorelisanim podacima. Simuliran je jedan klasifikator sa deset ulaznih promenljivih i jednom izlaznom koja može imati četiri vrednosti.

Tabela 11: Težinski faktori u različitim šemama glasanja kada su vrednosti klasa nekorelisane sa ulaznim podacima

Klasa	Tačnost	F1 metrika	Kapa-statistika
0	0.22	0.2	0
1	0.102	0.099	0
2	0.246	0.265	0
3	0.118	0.066	0

Kombinovanje vrednosti pročitanih sa senzora, između kojih postoji visok nivo interakcija može doneti bolje rezultate pri predviđanjima, ali će takođe povećati potrošnju energije u sistemu. Prema tome, ovakvo proširenje algoritma treba pažljivo koristiti. Ukoliko ovi senzori poseduju i jake marginalne efekte, rezultujuća globalna tačnost će možda samo minimalno biti poboljšana.

Pri simulacijama rada u realnom okruženju sistem takođe pokazuje zadovoljavajuće performanse (uz opasku da nivo neophodnih performansi da bi sistem bio upotrebljiv zavisi od konkretne aplikacije i projektnih zahteva). Ukoliko je izraženo prisustvo šuma, rad sistema se može poboljšati podizanjem minimalne granice iznad koje se glasovi uzimaju u obzir. U suprotnom, na rad sistema najviše utiče rezolucija senzorske mreže.

4. Distribuirane Kohonenove samoorganizujuće mape u bežičnim senzorskim mrežama

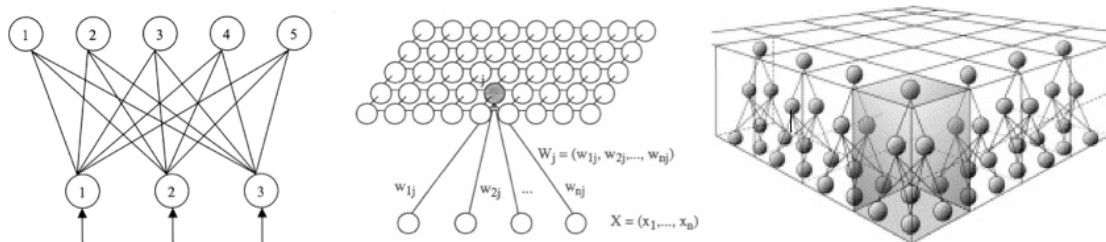
U poglavlju koje sledi biće predstavljen pristup kojim se Kohonenova neuralna mreža prilagođava upotrebi mašinskog učenja u bežičnoj senzorskoj mreži. Dati pristup sadrži elemente koji su originalna naučna kontribucija ove teze.

4.1. Kohonenove neuralne mreže

Kohonenove samoorganizujuće mape, koje se takođe nazivaju Kohonenove neuralne mreže, jesu tip algoritma za grupisanje. U osnovi, Kohonenove mreže su inspirisane mehanizmima koji se koriste u cerebralnom korteksu pri organizaciji podataka. Za razliku od najčešćeg tipa veštačkih neuralnih mreža (sa propagacijom greške unazad), Kohonenove neuralne mreže su nenadgledani algoritam, odnosno ne zahtevaju informaciju o tome kako podatke treba klasifikovati (drugim rečima, ne koriti informaciju o vrednosti klasne labele). Ovo ih čini veoma pogodnim za upotrebu u bežičnim senzorskim mrežama, pošto je eliminiše potreba da se podaci o klasama šalju do senzorskih čvorova pre faze treniranja.

Kohonenova mreža se sastoji od izvesnog broja čvorova, od kojih svaki sadrži skup težinskih faktora jednake dužine kao ulazni vektor podataka (odnosno, za svaki atribut u skupu podataka postoji po jedan težinski faktor, ove težine se još nazivaju i prototip vektor). Ovde nailazimo na jedan terminološki problem: izraz *čvor* se nezavisno koristi u domenima bežičnih senzorskih mreža i mašinskog učenja pomoću veštačkih neuralnih mreža, pri čemu ova dva pojma nisu ekvivalentna (u senzorskoj mreži čvor je fizički uređaj u sistemu, dok je u neuralnoj to opisana struktura sa težinskim faktorima). Da bismo izbegli zabunu u tekstu koji sledi, čvorove koji su vezani za koncept iz neuralnih mreža nazvaćemo *neuronima*, dok ćemo za koncept vezan za senzorsku mrežu uvek

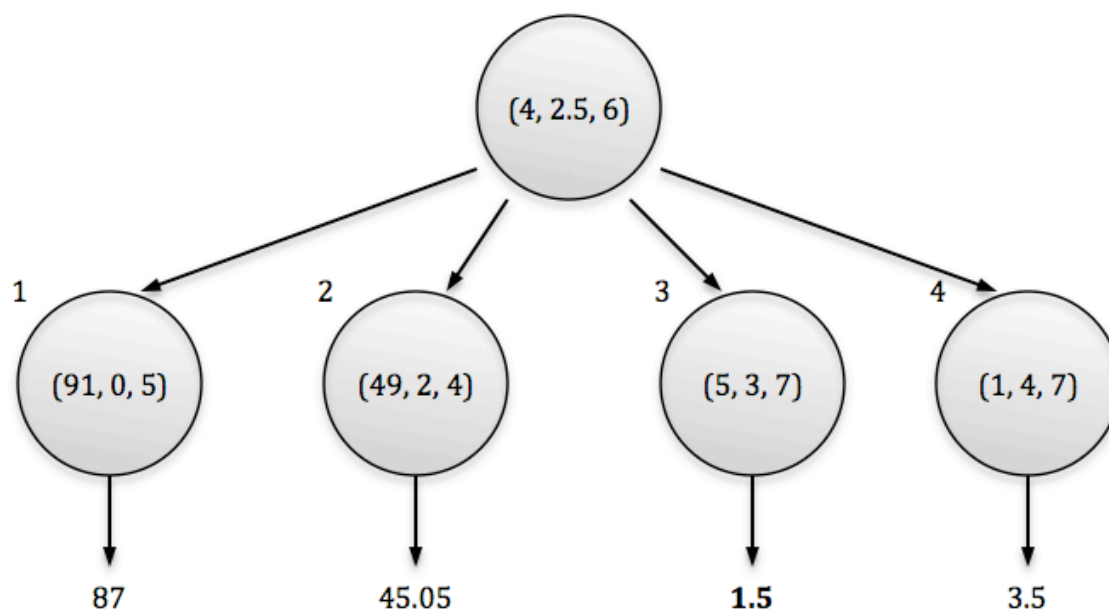
koristiti izraz *senzorski čvor*. Na Slici 40 su prikazane tri česte topologije koje se koriste pri formiranju Kohonenovih mreža. Najjednostavnija je jednodimenzionalna topologija, koja je ujedno i jedina za koju je dokazana konvergencija. Međutim, najčešće se koristi dvodimenzionalna verzija, pošto može da opiše znatno složenije skupove podataka, dok empirijski obično konvergira. Trodimenzionalne topologije su relativno slabo istražene zbog visoke kompleksnosti.



Slika 40. Primeri topologija Kohonenovih mreža a) jednodimenzionalna, b) dvodimenzionalna, c) trodimenzionalna

U toku rada Kohonenove mreže uzima se po jedna instanca podataka. Svaki neuron određuje Euklidsko rastojanje između svog trenutnog vektora težinskih faktora i vrednosti date instance podataka. Neuron za koji je ovo rastojanje najmanje od svih u mreži se proglašava za *pobednika*. Instanca podataka je najbliža grupi instanci za koje je isti pobednik isti neuron. Samoorganizujuća mapa se može interpretirati kao skup neurona, od kojih svaki predstavlja po jednu grupu podataka, pri čemu podatak pripada onoj grupi koja je predstavljena čvorom pobednikom. Međutim, samoorganizujuća mapa se može posmatrati i kao struktura koja predstavlja manji broj grupa podataka, tako što se uzme da po nekoliko bliskih neurona predstavlja jednu grupu (kao što će uskoro biti obrazloženo, procedura treniranja Kohonenove samoorganizujuće mape dovodi do toga da se neuroni koji imaju slične težinske faktore nalaze blizu u topologiji mape). Primer je dat na Slici 41. Ulazni podatak X će biti prosleđen na sve neurone u mreži (u ovom slučaju to su četiri čvora, realne samoorganizujuće mape su znatno veće) i svaki neuron će odrediti euklidsko

odstojanje tog podatka od svog težinskog vektora. Pobednik je čvor označen brojem 3.



Slika 41. Primer rada Kohonenove samoorganizujuće mape

4.2. Treniranje Kohonenove neuralne mreže

Faza treniranja Kohonenove neuralne mreže počinje tako što se svi težinski faktori u svim neuronima inicijalizuju sa nasumično odbranom malom vrednošću. Potom se za sve podatke koji su na raspolaganju u skupu podataka za treniranje određuje neuron pobednik, na isti način kao što je opisano. Pri ovome izračunavanju svi neuroni osvežavaju vrednosti svojih težinskih faktora, prema sledećoj formuli:

$$w = w + \alpha \cdot \eta(\text{pobednik}) \cdot (w - x) \quad (7)$$

gde w označava vektor težinskih faktora u datom neuronu, x predstavlja podatak iz skupa podataka koji se trenutno obrađuje, α je parametar koji se naziva stopa učenja, a η je funkcija susednosti kojom se obezbeđuje da čvorovi koji su topološki blizu imaju slične vektore težinskih faktora (praktično, ovo je funkcija

čija vrednost govori koliko su dva čvora blizu u topologiji mape). Funkcija susedstva kao argument uzima rastojanje između datog čvora i čvora koji je označen kao pobednik. Na primer, na Slici 41 rastojanje između čvorova označenih brojevima 1 i 3 je 2, između čvorova 2 i 3 je 1, itd. Kod višedimenzionalnih struktura se uzima Euklidsko odstojanje. Uloga funkcije susedstva je da ovo odstojanje prevedu u formu koja će obezbediti stabilno učenje u mreži. Često izbor je Gausova funkcija.

4.3. Primena Kohonenovih samoorganizujućih mapa u bežičnim senzorskim mrežama

Mogućnosti prilagođavanja Kohonenovih samoorganizujućih mapa primenama u bežičnim senzorskim mrežama su analizirane u nekoliko publikacija [Bicocchi, 2007a], [Bicocchi, 2007b], [Catterall, 2003], [Yun, 2007].

Istraživanje koje je najbliže pristupu koji će biti opisan u nastavku su predložili Kateral i grupa autora [Catterall, 2003]. Oni mapiraju svaki neuron u Kohonenovoj neuralnoj mreži na jedan čvor u bežičnoj senzorskoj mreži. Vektor ulaznih podataka neurona je vektor očitanih vrednosti sa senzora senzorskog čvora. Dakle, u pitanju je horizontalna distribucija podataka. Ovakva struktura dovodi do toga da se u jednom trenutku ulazi na različitim neuronima razlikuju, što nije slučaj kod običnih samoorganizujućih mapa. Rastojanja između neurona se određuju kao fizičke distance između čvorova bežične senzorske mreže na kojima se nalaze, umesto logičkih distanci u samoorganizujućoj mapi. Jun i grupa autora su dodatno proširili ovaj pristup, tako da se kod računanja odstojanja uzima u obzir i jačina primljenog signala, kako bi se fizičke distance bolje odredile [Yun, 2007].

Za razliku od navedenih primera, u tekstu koji sledi se predlaže upotreba logičkih odstojanja u samoorganizujućoj mapi. Drugim rečima, usvaja se

pristup iz klasičnih samoorganizujućih mapa kod određivanja rastojanja između neurona (kao što je opisano u uvodnom delu). Takođe, uvodi se modifikacija algoritma koja omogućava da se komunikacija odvija samo između senzorskih čvorova koji sadrže neurone koji su susedni u topologiji samoorganizujuće mape (dok postojeća rešenja zahtevaju da svi senzorski čvorovi razmene podatke, ili da se rad sistema u potpunosti centralizuje). Konačno, uvodi se i unapređeno rešenje, koje omogućava da se uzmu u obzir dva različita tipa varijacija u očitavanjima sa senzora: varijacije usled promena u okruženju (to su promene koje izazivaju različiti događaji koji se žele detektovati i opisati) i promene koje nastaju usled propagacije različitih signala (čvor koji je udaljen od izvora promene u okruženju može imati drugačija očitavanja od onoga koji je blizu).

4.4. Opis predloženog rešenja

U nastavku će biti dat detaljniji opis predloženih rešenja. Prvo će biti izložen osnovni algoritam upotrebe Kohonenove neuralne mreže koji može da služi ili za lokalizaciju ili za karakterizaciju događaja u oblasti koja se nadgleda bežičnom senzorskom mrežom. Potom će biti opisano i poboljšanje koje omogućava da se istovremeno vrše i lokalizacija i karakterizacija. Poboljšana verzija algoritma je posebno bitna za slučajeve kada efekti koji su izazvani promenom u okruženju opadaju sa rastojanjem od mesta gde je ta promena nastala (recimo povećanje temperature usled eksplozije), tako da je promena u intenzitetu efekata značajna na odstojanjima između senzorskih čvorova.

4.4.1 Osnovni pristup za upotrebu Kohonenove samoorganizujuće mape u bežičnoj senzorskoj mreži

U osnovnom pristupu neuroni Kohonenove neuralne mreže se vezuju za čvorove bežične senzorske mreže tako što svakome senzorskom čvoru odgovara po jedan neuron. Ovakav pristup liči na onaj opisan u [Catterall, 2003], pa ćemo ponoviti tamo definisan skup razlika u odnosu na tradicionalne samoorganizujuće mape, uz unete potrebne razlike:

- Elementi samoorganizujuće mape su predstavljeni senzorskim čvorovima; svaki senzorski čvor sadrži prototip vektor (lokalni vektor težinskih faktora) koji po strukturi odgovara vektoru očitavanja sa lokalnih senzora.
- Topologija samoorganizujuće mape je definisana topologijom bežične senzorske mreže. Za razliku od rešenja opisanog u referenci [Catterall, 2003] rastojanja u algoritmu samoorganizujuće mape su logička rastojanja u topologiji, a ne fizička rastojanja u prostoru između senzorskih čvorova.
- Ulazni vektor svakog neurona u samoorganizujućoj mapi uzima vrednosti koje se očitavaju sa senzora na datom senzorskom čvoru.

U toku rada svaki senzorski čvor uzima očitavanja sa svojih senzora i izračunava Euklidsko odstojanje od svog lokalnog prototip vektora. Kao u klasičnoj Kohonenovoj neuralnoj mreži, čvor koji izračuna najmanje odstojanje se proglašava pobednikom. Pri osvežavanju vrednosti lokalnih prototip vektora svakome neuronu je potrebna informacija o tačnoj lokaciji neurona pobednika, kako bi mogao da odredi vrednost funkcije susedstva. Da bi se postiglo da svi neuroni dobiju ovakav podatak, bilo bi potrebno da svi čvorovi u senzorskoj mreži razmene podatke, svaki sa svakim (ukoliko je poznata specifična topologija mreža, ovaj uslov je moguće optimizovati i olabaviti, ali bi se

svejedno zahtevala značajna količina komunikacije). Kako bi se ovakva komunikacija izbegla, predlaže se novo rešenje za približno izračunavanje rastojanja od nekog neurona do neurona pobednika.

Pretpostavimo da za trenutni podatak u samoorganizujućoj mapi postoji neuron čiji prototip vektor savršeno odgovara vrednosti podatka, odnosno da postoji čvor koji ima najbolji mogući odziv na trenutni podatak (u praktičnom smislu da je neki čvor izračunao Euklidsko odstojanje između podatka i prototip vektora koje je jednako nuli). Stvarno postojanje ovakvog neurona može biti malo verovatno, te se ovakav hipotetički neuron možda neće u potpunosti poklapati sa nekim realnim neuronom u mreži; dakle samo pretpostavljamo da neuron postoji i da se nalazi negde unutar samoorganizujuće mape. S obzirom na to da se u samoorganizujućoj mapi neuroni koji imaju slične prototip vektore (odnosno jake odzive na slične podatke) nalaze topološki blizu, ovakav „savršeni pobednik“, da postoji, nalazio bi se u blizini stvarnog neurona pobednika za dati podatak.

Umesto da računa odstojanje u odnosu na lokaciju stvarnog pobednika, svaki neuron će izračunati vrednost funkcije susedstva na osnovu procenjenog odstojanja od uvedenog hipotetičkog savršenog pobednika. Jednačina (7) prema tome postaje:

$$w = w + \alpha * \eta (\text{savršeni_pobednik}) * (w - x) \quad (8)$$

pri čemu w , α , η , x imaju isto značenje kao i u jednačini 7.

Ostaje pitanje kako sada odrediti odstojanje do hipotetičkog savršenog pobednika? Neka svaki senzorski čvor razmeni podatke od odstojanja koje je izračunato u sklopu lokalnog neurona sa njegovim neposrednim susedima u samoorganizujućoj mapi. Ovime se uvodi komunikacija između bliskih čvorova, koji su u neposrednom međusobnom dometu. Treba napomenuti da čvor neće komunicirati sa svim neposrednim susedima u senzorskoj mreži, već samo sa

onima koji ujedno sadrže i neurone koji su susedni u samoorganizujućoj mapi. Ukoliko pretpostavimo dvodimenzionalnu grid topologiju (Slika 40b), to znači da će svaki senzorski čvor razmeniti podatke sa najviše četiri susedna čvora, bez obzira na veličinu senzorske mreže (dok će čvorovi na ivici mreže komunicirati sa manje od četiri čvora).

U Kohonenovim mrežama važi da što su dva neurona bliža to sličnije reaguju na iste ulazne podatke. Prema tome, neuroni koji su topološki blizu savršenom pobedniku će izračunati manje distance od onih koji su udaljeniji. Drugim rečima, ukoliko bi se kretali kroz mrežu, izračunate distance bi se smanjivale od neurona do neurona kako se približavamo savršenom pobedniku. Ukoliko je poznata vrednost na nekoliko susednih neurona, trend opadanja distanci je moguće pronaći, a potom ekstrapolirati na celu mrežu. Dakle, predloženo rešenje je da svaki čvor metodom linearne regresije lokalno odredi funkciju koja predviđa vrednost rastojanja između trenutnog podatka i prototip vektora nekog čvora. Ulazni parametar je rastojanje između datog čvora i čvora za koji se određuje odstojanje. Postavljanje vrednosti ove funkcije na nulu omogućava da se inverzno odredi rastojanje između trenutnog neurona i savršenog pobednika. Linearna aproksimacija se može učiniti previše jakom pretpostavkom, ali kao što se vidi iz simulacionih analiza, daje dosta dobru procenu, a pri tome je jednostavna za određivanje. Kao funkcija susedstva se može koristiti Gausova funkcija.

Prema formuli 8, za određivanje vrednosti kojima se osvežavaju lokalni prototip vektori, preostaje da se postavi vrednost parametara α , kojim se utiče na brzinu učenja u mreži. Ukoliko se usvoji viša vrednost, sistem će učiti brže i može se očekivati da do konvergencije dođe u kraćem vremenskom periodu. Takođe, sa višom vrednošću ovog parametra može se očekivati da će sistem brže moći da se prilagodi promenama u okruženju koje dovode do promena u zakonitostima u podacima [Gruenwald, 2007]. Sa druge strane, ukoliko se ovaj parametar postavi na previše visoku vrednost, može se dogoditi da do konvergencije ne

dode, kao i da dode do konstantnog preteranog treniranja na osnovu zakonitosti u poslednjih par viđenih podataka.

Prethodne empirijske analize [Catterall, 2003] su pokazale da su u sličnim sistemima, sa Kohonenovom samoorganizujućom mapom u bežičnoj senzorskoj mreži, optimalne vrednosti parametara α oko 0.4. U našim simulacijama je kao optimalna određena nešto manja vrednost, oko 0.3. Treba napomenuti da je optimalna vrednost ovog parametra u velikoj meri zavisna od aplikacije i konkretnog skupa podataka sa kojim se radi. Prema tome, brojke koje su ovde date treba posmatrati kao početne tačke, od kojih se može krenuti pri optimizaciji.

4.4.2. Prošireni pristup za upotrebu Kohonenove samoorganizujuće mape u bežičnoj senzorskoj mreži

Kao motivaciju za razvoj ovog proširenog pristupa navešćemo sledeću tvrdnju: *U bežičnoj senzorskoj mreži kojom se nadgleda neka oblast postoje dva izvora varijacija u podacima koji se dobijaju sa senzora na čvorovima (pored eventualnog prisustva šuma). Prvi izvor varijacija su različiti događaji i stanja u okruženju koji utiču na veličine koje se prate sensorima (odgovor na pitanje: Šta se događa u oblasti koja se posmatra?). Drugi izvor varijacije je fizičko rastojanje između različitih senzorskih čvorova u bežičnoj senzorskoj mreži i tačnog mesta u posmatranoj oblasti na kojoj su se dati događaj ili promena dogodili (odgovor na pitanje: Gde tačno se u posmatranoj oblasti promena dogodila).*

U maloj senzorskoj mreži, u kojoj su svi senzorski čvorovi blizu jedan drugome, prvi naveden izvor varijacija će verovatno biti dominantan. U tom slučaju će samoorganizujuća mapa napravljena po opisanom osnovnom pristupu kategorizovati događaje prema njihovom tipu, odnosno fizičkim svojstvima (na primer, da li je u pitanju eksplozija ili udar groma). Međutim, ako je u pitanju velika senzorska mreža koja pokriva veću oblast, pri čemu su i rastojanja između čvorova veća, drugi izvor varijacija će igrati značajnu ulogu. Kao

posledica, samoorganizujuća mapa će postati osetljivija na rastojanje čvorova od izvora varijacije, odnosno počće da kategorizuju događaje prema mestu na kojem su se dogodili (pri čemu će čvor pobednik biti onaj koji je najbliži mestu na kojem se događaj desio). Prošireni pristup je razvijen za primenu u ovom drugom slučaju, sa većim senzorskim mrežama.

Krenimo od samoorganizujuće mape prilagođene bežičnim senzorskim mrežama, kao što je opisano u osnovnom pristupu. U ovako napravljenom sistemu samoorganizujuća mapa će reagovati na fizičku lokaciju događaja u oblasti koja se nadgleda. Neuron koji je proglašen za pobednika će biti onaj koji se nalazi na senzorskom čvoru koji je najbliži mestu događaja (u idealnom slučaju; u realnosti će to biti jedan od najbližih). Kako bismo dalje mogli da razlikujemo i tipove događaja (pošto je približna lokacija već poznata) možemo kreirati još jednu samoorganizujuću mapu. Ova druga mapa može u potpunosti biti smeštena na čvoru koji sadrži neuron koji je pobednik u prvoj mapi. Prvu mapu, kojom se određuje lokacija i koja je kreirana kao u osnovnom pristupu, ćemo nazvati *globalna mapa*, pošto je distribuirana kroz celu senzorsku mrežu. Drugu mapu, koja služi za karakterizaciju, i koja se nalazi na čvoru koji sadrži neuron koji je pobednik u globalnoj mapi ćemo nazvati *lokalna mapa*, pošto se nalazi na jednom čvoru.

U toku rada sistema se može desiti da svaki čvor bude, u nekom trenutku, proglašen za pobednika, pa prema tome svaki senzorski čvor uz jedan globalni neuron sadrži i po jednu lokalnu samoorganizujuću mapu, koju pravi, održava i koristi potpuno autonomno u odnosu na ostatak mreže. Pretpostavimo za početak da se lokalna mapa koristi i osvežava samo kada je globalni neuron na datom čvoru proglašen za pobednika. Ovakav pristup je opravdan, pošto su u tom slučaju očitavanja sa senzora „najčistija“, odnosno najbliže opisuju dešavanja u okruženju, bez uticaja propagacije efekata kroz prostor (na primer, slabljenja toplote koju je izazvala eksplozija kako se udaljavamo od mesta gde

se ona dogodila). Prema tome, lokalna mreža može da vrši kategorizaciju događaja samo po njihovom tipovima.

Međutim, ukoliko bi se vrednosti lokalne mape osvežavale samo u situacijama u kojima je globalni neuron na odgovarajućem senzorskom čvoru proglašen za pobjednika, došlo bi do upotrebe veoma malog procenta raspoloživih podataka. Na primer, neka senzorska mreža sadrži 100 senzorskih čvorova. Ukoliko bi svaki od 100 neurona u globalnoj mapi jednako često bivao proglašen za pobjednika, to bi značilo da se svaka lokalna mapa trenira sa stotim delom raspoloživih podataka. Time bi se vreme do konvergencije praktično povećalo 100 puta.

Da bismo rešili ovaj problem, predlažemo dodatno proširenje algoritma, pri čemu usvajamo sledeću tvrdnju: *Čvor koji je najbliži mestu događaja ima „najjasniji pogled“ na događaj koji se desio. Međutim, i drugi senzorski čvorovi koji su u blizini mogu imati „relativno jasan pogled“ na događaj, u zavisnosti od toga koliko su udaljeni.* Prema tome, lokalne mape se mogu osvežavati i u situacijama kada globalni neuron na odgovarajućem senzorskom čvoru nije proglašen za pobjednika.

Kako bismo uzeli u obzir činjenicu da će podaci očitani sa senzora na čvorovima koji su bliži njihovom izvoru biti manje izmenjeni propagacijom kroz prostor (a time i bolje opisivati događaj koji ih je izazvao), uvodimo dodatni parametar kojim se kontroliše brzina učenja lokalne mape, a koji zavisi od udaljenosti senzorskog čvora od mesta događaja. Podatak o udaljenosti čvora od mesta događaja je u trenutku osvežavanja lokalne mape već poznat - u formi distance koju je izračunao globalni neuron da datom čvoru. Prema tome, formula za računanje vrednosti kojom se osvežava lokalna mapa postaje:

$$w = w + \alpha * \eta (\text{lokalni pobjednik}) * (w - x) * \beta \quad (9)$$

pri čemu w , α , η , x imaju isto značenje kao i u jednačinama 7 i 8, a novi parametar β predstavlja distancu koju je izračunao globalni neuron.

4.5. Analiza rada osnovne verzije sistema na osnovu simulacije

Kako bi se izvršila validacija opisanih rešenja i njihove efikasnosti, urađeni su skup simulacija i analiza dobijenih rezultata. Kao i kod prikaza teorijskih osnova predloženih pristupa, prvo su dati opis i rezultati za osnovnu verziju algoritma, a potom i za proširenu verziju.

4.5.1. Metodologija

Simuliran je rad bežične senzorske mreže koja se sastoji od 100 senzorskih čvorova. Senzorski čvorovi su postavljeni u regularnu kvadratnu grid topologiju (10x10 čvorova). Svaki čvor sadrži tačno jedan senzor.

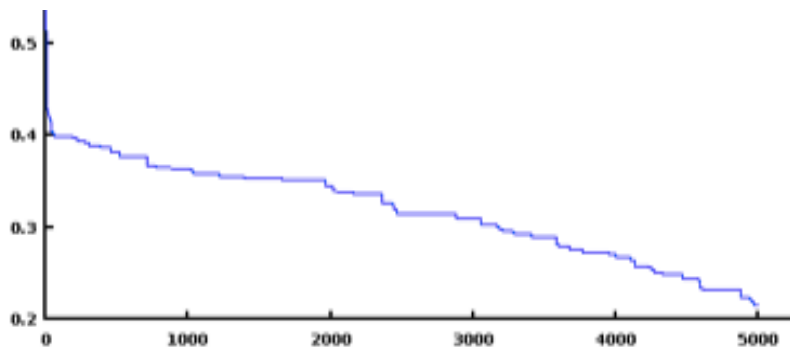
Okruženje u kojem je smeštena senzorska mreža je simulirano tako što su kao vrednosti senzorskih merenja izvlačeni brojevi iz Gausove distribucije fiksirane srednje vrednosti i varijanse. Lokacija na kojoj se događaj desio je nasumično odabirana u svakom koraku simulacije. Iz ove tačke je simulirana propagacija efekata događaja (čija je maksimalna vrednost, u datoj tački izvučena iz Gausove raspodele) sa linearnim slabljenjem. Kako bi se simuliralo prisustvo šuma u svakoj tački je dodati i beli aditivni šum, maksimalne vrednosti 0.15 (15% maksimalne vrednosti efekta koji događaj može da izazove). Gledano po koracima, svaka iteracija simulacione procedure se sastoji od sledećih koraka:

- Simuliraj događaj izvlačenjem amplitude iz Gausove raspodele i lokacije iz dvodimenzionalne uniformne distribucije;
- Izračunaj propagaciju efekata događaja za celokupnu oblast u kojoj se bežična senzorska mreža nalazi;
- Dodaj šum u svakoj tački;
- Za svaki senzorski čvor uzmi očitavanja sa senzora;

- Izračunaj izlaze iz svih neurona u mreži i izračunaj vrednosti kojima se osvežavaju prototip vektori, procedurom koja je opisan kao osnovni mehanizam rada.

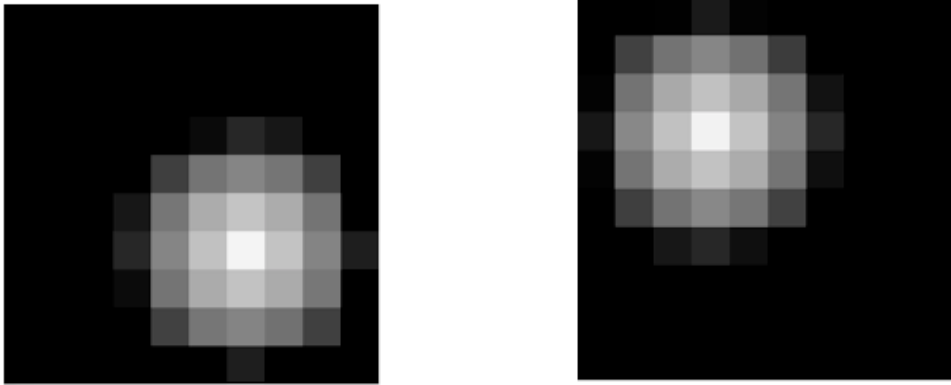
Prikazani rezultati su prosečne vrednosti dobijene iz 100 ponavljanja simulacione procedure, svake sa po 5.000 iteracija.

Na Slici 42 je prikazana greška (izlazno odstojanje iz čvora koji je označen kao pobednik) u toku simulacije. Posle prvih par iteracija dolazi do naglog pada greške, koja potom nastavlja lagano da opada u nastavku simulacije. Takav obrazac ponašanja je karakterističan za samoorganizujuće mape, što znači da rad sistema ne odstupa značajno od klasičnih rešenja.



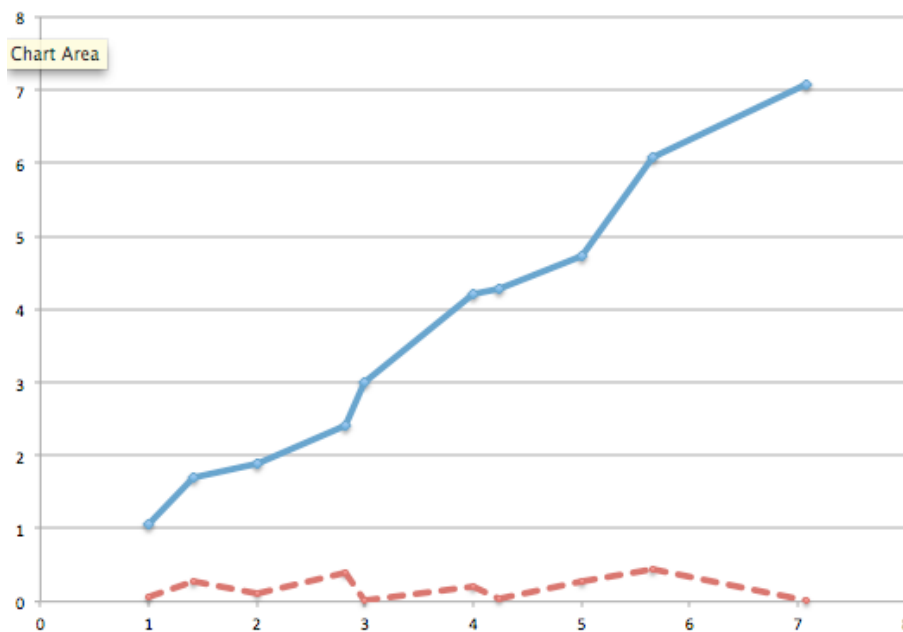
Slika 42: Izlazne vrednosti iz neurona koji je proglašen za pobednika, u toku simulacije rada sistema

Na Slici 43 je prikazan izlaz iz samoorganizujuće mape za dva simulirana događaja. Čvorovi sa nižim izlaznim vrednostima (oni koji su izračunali da su blizu događaja) su dati svetlijom bojom. U oba slučaja mapa je ispravno odredila mesto događaja.



Slika 43: Izlazne vrednosti iz samoorganizujuće mape za dva simulirana događaja. Čvorovi sa nižim izlaznim vrednostima (oni koji su izračunali da su blizu događaja) su dati svetlijom bojom. U oba slučaja mapa je ispravno odredila mesto događaja.

Na Slici 44 su prikazani efekti aproksimacije upotrebom procenjenog odstojanja od „savršenog pobednika“, umesto upotrebe stvarnog pobednika i stvarnog odstojanja od njega. U treniranom sistemu (posle 5.000 iteracija rada sistema) simuliran je jedan događaj i izračunata su odstojanja na osnovu procedure sa savršenim pobednikom. Grafikon prikazuje ova odstojanja u funkciji od odstojanja do stvarnog pobednika, i grešku koja je pri tome napravljena. Sa slike se vidi da su aproksimativna odstojanja veoma blizu stvarnim.



Slika 44: Procenjene vrednosti (puna linija) odstojanja do savršenog pobednika i načinjena greška (isprekidana linija)

4.6. Analiza rada proširene verzije sistema na osnovu simulacije

Bežična senzorska mreža iste veličine i topologije kao pri simulacijama osnovne verzije sistema je korišćena i za simulaciju sistema koji radi na osnovu proširenog algoritma (100 senzorskih čvorova, 10x10 kvadratna grid topologija).

Pošto je prošireni pristup namenjen omogućavanju kategorizacije događaja prema njihovom tipu, bilo je neophodno uvesti nekoliko različitih tipova događaja (u osnovnom pristupu je jedino menjana lokacija na kojoj se događaj desio). To je urađeno tako što je uvedeno da je stanje okruženja u kojem se mreža nalazi definisana sa dva parametra, *promenljiva 1* i *promenljiva 2*. Svaki senzorski čvor je opremljen sa tačno dva senzora, po jednim sa kojim može da detektuje veličine vezane za dva parametra okruženja. Pri svakoj iteraciji simulacije, svaki od dva parametara okruženja se nezavisno izvlači iz jedne od dve moguće Gausove raspodele (koje imaju različite srednje vrednosti i varijanse). Stanje okruženja (odnosno tip događaja koji se desio) je definisan kao jedna od četiri moguće vrednosti, u zavisnosti od toga iz kojih su raspodela izvučeni parametri, što je prikazano u Tabeli 12.

Tabela 12: Vrednosti promenljivih koje definišu stanje u okruženju, odnosno tip događaja koji je simuliran

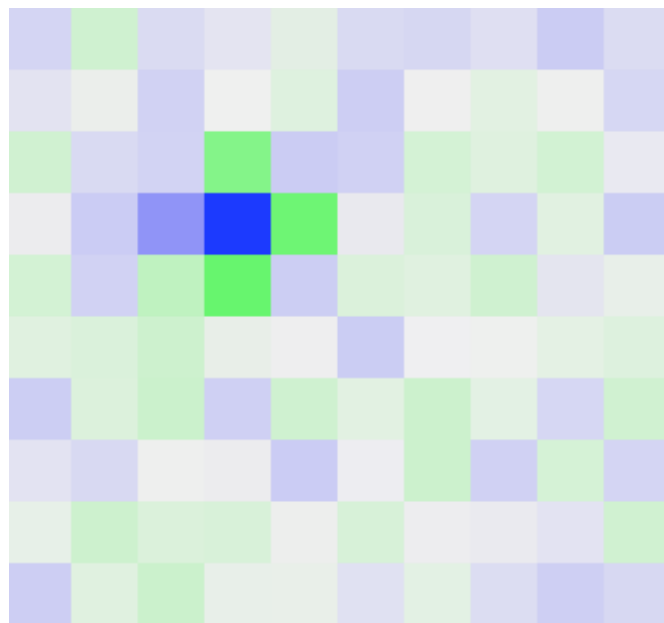
Događaj	Promenljiva1	Promenljiva2
1	Gaus1	Gaus1
2	Gaus1	Gass2
3	Gaus2	Gaus1
4	Gaus2	Gaus2

Lokalne samoorganizujuće mape su formirane sa jednodimenzionalnom topologijom, i svaka sadrži po četiri neurona. Šum je dodavan na isti način kao što je opisano za simulacije osnovne verzije algoritma. Po koracima svaka iteracija simulacione procedure se sastoji od:

- Simuliraj tip događaja, izvlačeći nasumice iz skupa {1,2,3,4};

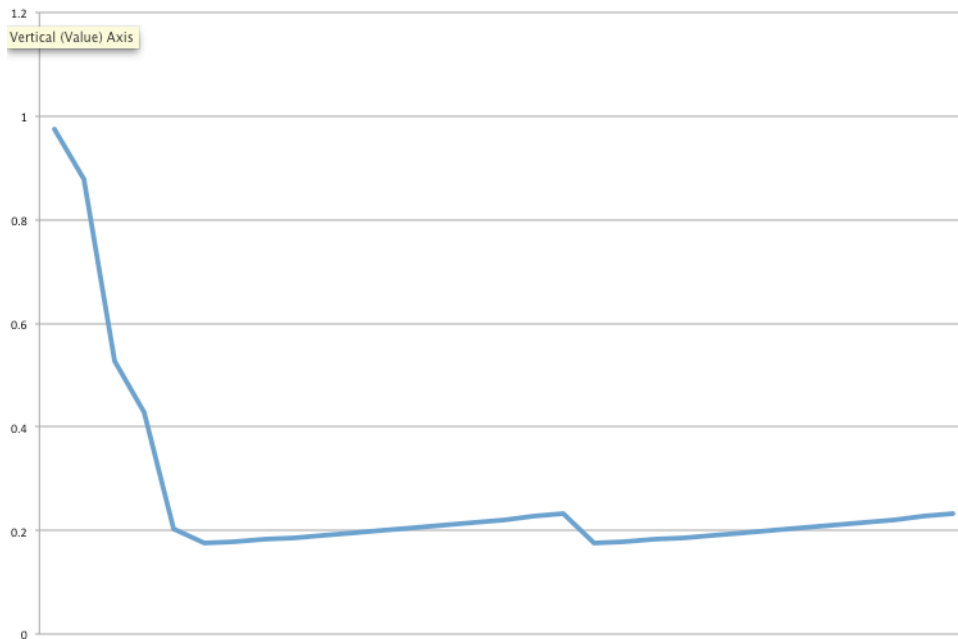
- Simuliraj događaj izvlačenjem amplitude parametara iz odgovarajućih Gausovih raspodela (Tabela 12) i lokacije iz dvodimenzionalne uniformne distribucije;
- Izračunaj propagaciju efekata događaja za celokupnu oblast u kojoj se bežična senzorska mreža nalazi;
- Dodaj šum u svakoj tački;
- Za svaki senzorski čvor uzmi očitavanja sa senzora;
- Izračunaj izlaze iz svih globalnih neurona u mreži i izračunaj vrednosti kojima se osvežavaju prototip vektori, procedurom koja je opisana kao osnovni mehanizam rada;
- Izračunaj izlaze iz lokalnih mapa i izračunaj vrednosti kojima se osvežavaju prototip vektori, procedurom koja je opisana kao prošireni mehanizam rada.

Na Slici 45 su prikazani izlazi iz sistema za jedan simulirani događaj na kraju procedure simulacije (posle 5.000 iteracija). Bojom je predstavljen tip događaja, a nijansom distanca čvora koja je procenjena na globalnom nivou (s tim da su tamnije nijanse niže vrednosti izračunate distance, odnosno čvorovi koji su tamniji su bliži lokaciji događaja).



Slika 45: Izlazne vrednosti iz sistema za jedan simulirani događaj, na kraju procedure simulacije. Nijanse označavaju odluku iz globalne mape (tamnije nijanse označavaju da je globalni neuron bliži događaju), a boje označavaju tip događaja (lokalnog pobednika na datom neuronu)

Na Slici 46 je prikazana greška (izlazno odstojanje iz čvora koji je označen kao pobednik) u toku simulacije. Posle prvih par iteracija dolazi do naglog pada greške, koja potom nastavlja lagano da opada u nastavku simulacije. Takav obrazac ponašanja je karakterističan za samoorganizujuće mape, što znači da rad sistema ne odstupa značajno od klasičnih rešenja. U ovom slučaju procedura treniranja je nešto sporija nego kod osnovnog rešenja.



Slika 46: Izlazne vrednosti iz neurona koji je proglašen za pobjednika, u toku simulacije rada sistema. Vrednosti su date za neuron koji je proglašen za lokalnog pobjednika, na senzorskom čvoru koji sadrži neuron koji je pobjednik u globalnoj mapi

4.7. Diskusija predloženih rešenja za upotrebu Kohonenove samoorganizujuće mape u bežičnoj senzorskoj mreži

Prikazana su dva rešenja za upotrebu Kohonenove samoorganizujuće mape u bežičnim senzorskim mrežama. Prvi, jednostavniji pristup, koji smo označili kao *osnovni algoritam* je u stanju da vrši karakterizaciju događaja u malim, odnosno lokalizaciju događaja u velikim i prostranim senzorskim mrežama. U toku rada algoritma zahtevi za komunikacijom su svedeni na razmenu male količine podatka (jednog broja) sa nekolicinom okolnih čvorova (tačan broj zavisi od topologije samoorganizujuće mape koja se želi postići; u slučaju dvodimenzionalne mreže do su četiri okolna čvora).

Prošireni pristup ima sposobnost da vrši i lokalizaciju i karakterizaciju događaja u oblasti koju pokriva velika i prostrana senzorska mreža. Proširenje se sastoji od uvođenja još jednog, lokalnog, nivoa mape, na svakom senzorskom čvoru. Ove lokalne mape se treniraju i koriste nezavisno jedna od druge, pa ne zahtevaju nikakvu komunikaciju u toku rada. Treba napomenuti da kod ovih

lokalnih mapa ne postoji garancija da će različiti čvorovi iste događaje mapirati na topološki iste neurone. Do ovoga dolazi zato što su na početku svi težinski faktori postavljeni na slučajno odabrane vrednosti, koje se nezavisno postavljaju na svakom čvoru. Pomenuta činjenica ne utiče na ispravnost rada sistema, ali se mora uzeti u obzir pri interpretaciji rezultata dobijenih iz lokalnih mapa.

Prošireni algoritam se bazira na pretpostavci da su senzorski čvorovi koji se nalaze najbliže lokaciji na kojoj se desio posmatrani događaj oni čija očitavanja daju najviše informacija o tipu i karakteru tog događaja. Iako je ovo često slučaj, u nekim situacijama data pretpostavka nije zadovoljena (na primer, različiti efekti kao što su zvuk i temperatura se različito prenose kroz okruženje, pa je za izvesne distance lakše proceniti šta se zaista desilo). Za ovakve situacije su potrebne dodatne modifikacije algoritma (na primer u vidu drugačijeg računanja β parametra).

U pogledu računске kompleksnosti oba pristupa su relativno jednostavna i pogodna za upotrebu u bežičnim senzorskim mrežama čiji su čvorovi ograničenog kapaciteta. Osnovni pristup zahteva da samo jedan težinski prototip vektor bude smešten u memoriji, dok je kompleksnost potrebnih izračunavanja mala i zavisi od broja senzora na čvoru, kao $O(n)$ (pri čemu je n broj senzora na čvoru). Prošireni pristup zahteva da senzorski čvor sadrži i lokalnu samoorganizujuću mapu. U ovom slučaju je kompleksnost (i u pogledu potrebne memorije i potrebnog izračunavanja) $O(m \times n)$, gde je n broj senzora na svakom čvoru, a m broj neurona u lokalnoj samoorganizujućoj mapi (što zavisi od broja različitih događaja koji se žele karakterizovati). Kako se za brojeve m i n može očekivati da budu relativno mali, i ovaj prošireni pristup je pogodan da većinu aplikacija bežičnih senzorskih mreža.

Opisani algoritmi se mogu dodatno proširiti kako bi se iskoristilo i znanje dostupno kroz ručno obeležavanje podataka (čime bi se algoritam učinio nadgledanim, ili polunadgledanim). Labele bi se mogle pridružiti nekim od podataka koji su očitani sa senzora i na taj način bi se označio tip događaja koji

je vezan za data očitavanja. Time bi se fizička značenja dešavanja u okruženju mogla dodeliti odgovarajućim oblastima u mapi koja na njih reaguje. Na ovaj način bi se rešio i pomenuti problem kod proširenog rešenja kada različite lokalne mape mapiraju iste događaje na svoje različite delove. Još jedno moguće unapređenje bi se odnosilo na uvođenje većinske logike, slično rešenju koje je opisano pri vertikalno distribuiranoj klasifikaciji. Umesto da se gleda samo izlaz iz lokalne mape čvora sa najboljim globalnim odzivom, odluke bi mogle da se donose putem (težinskog) glasanja većeg broja čvorova, pri čemu bi težinski koeficijenti mogli biti (ili sadržati u sebi) upravo globalni odziv čvora. Ovakav pristup bi mogao biti alternativno rešenje za problem koji se javlja ukoliko informacije postaju jasnije ako se posmatraju sa neke razdaljine od mesta događaja.

5. Zaključak

Prvobitne vizije razvoja bežičnih senzorskih mreža su predviđale da su senzorski čvorovi uređaji čiji računarski kapaciteti neće značajno rasti sa vremenom. Očekivalo se, naime, da se tehnološki napredak koji omogućava Murov zakon iskoristi kako bi se razvijali sve minijaturniji i jeftiniji uređaji. Ovakva predviđanja se međutim nisu ostvarila (bar ne u potpunosti). Na primer uzмимо Mica2 senzorski čvor iz 2002. godine i uporedimo njegov osmobaritni procesor koji radi na učestanosti od 4 megaherca sa Louts čvorom koji ima tridesetdvobaritni procesor koji radi na punih 100 megaherca. Veličina ova dva senzorska čvora je slična.

U tom svetlu se nameće zaključak da ulaganje značajnog napora kako bi se smanjile računске kompleksnosti programa koji se izvršavaju na senzorskim čvorovima nije stavka sa visokim prioritetom. Naravno, ovakav zaključak ne znači da se računска kompleksnost može u potpunosti zanemariti, pogotovo kod algoritama mašinskog učenja koje se može oslanjati na veoma intenzivna izračunavanja, na primer kvadratno programiranje ili invertovanja velikih matrica.

Sa druge strane, energija koja je na raspolaganju jednom senzorskom čvoru ne raste ni približno brzo kao računски kapaciteti senzorskih čvorova. Pristupi kao što je prikupljanje energije iz okoline [Gilbert, 2008] su predlagani, ali praktična primenljivost i komercijalna upotrebljivost ovakvih ideja nije sasvim jasna. Pošto je komunikacija veliki potrošač u radu bežičnih senzorskih mreža, njeno smanjivanje je svakako prioritet.

Veličina ostvarljivih senzorskih mreža će takođe vrlo verovatno rasti. Prema tome, skalabilnost rešenja se nameće kao visok prioritet kod razvoja novih algoritama. Potrebno je uzeti u obzir probleme kao što su kašnjenja u mreži, rast

količine komunikacije sa rastom mreže, kao i opterećenje na eventualnim centralnim lokacijama u sistemu.

Jedan trend koji uzima sve više maha se odnosi na interoperabilnost i deljenje informacija između većeg broja različitih senzorskih mreža, kao i omogućavanje pristupa većem broju raznovrsnih korisnika. U ovakvim scenarijima korišćenja podataka iz bežičnih senzorskih mreža akcenat je stavljen na bezbednost, očuvanje privatnosti i poverenje (garantovanje verodostojnosti informacija). Kao što se može videti iz teksta koji prethodi, upotreba algoritama distribuiranog mašinskog učenja može ponuditi rešenja za neke od problema u ovom domenu. Čak se može očekivati da će se u ovom domenu pronaći neki od osnovnih motivacionih faktora za širu upotrebu mašinskog učenja u bežičnim senzorskim mrežama.

Jedna interesantna opservacija se može napraviti na osnovu sadržaja pregledanih radova koji predlažu različite algoritme mašinskog učenja za bežične senzorske mreže. Naime, veoma je očigledna razlika u istraživanjima koja se izvode u matematičkim i informatičkim krugovima i onima koja dolaze iz laboratorija koje su primarno usmerene ka bežičnim senzorskim mrežama. Ovakva situacija je donekle i očekivana. Međutim, dati trend ima veoma duboke posledice po razumevanje kako bi algoritam mogao da bude upotrebljen i pretpostavkama koje je bezbedno napraviti sa jedne strane i matematičkih tehnika i ideja koje se mogu efikasno upotrebiti sa druge strane. Iz ovakvog stanja se vidi da bi obe istraživačke zajednice imale koristi od međusobne multidisciplinarne saradnje. Dalji napredak u ovom smislu bi se mogao ostvariti napuštanjem slojevite strukture u istraživanju i razvoju.

Primer se može videti na već opisanoj situaciji u kojoj se pokušava razviti energetski algoritam tako što se izbacuje što je moguće više komunikacije. Dosta napora se može uložiti u razvoj tehnika kojima se izbegava potreba da se neki podatak prenese sa jednog na drugi čvor. Dati podatak se međutim, na kraju, ipak mora preneti na neki treći čvor, tako da se drugi pomenuti čvor nalazi na

putanji između prvog i trećeg čvora. Time se eliminiše sva ušteda u komunikaciji. Štaviše, način na koji se vrši komunikacija između udaljenih čvorova u sistemima bežičnih senzorskih mreža dovodi do toga da podaci prolaze kroz veći broj čvorova nego što to logika aplikacije zahteva. Dakle, ukoliko bi u trenutku razvoja algoritma bilo poznato kako će se vršiti rutiranje u sistemu, učenje bi se moglo prilagoditi tako da se iskoriste svi podaci koji su raspoloživi na čvoru.

Nešto drugačiji problem se može videti kod pristupa distribuiranom mašinskom učenju koje se oslanjaju na grupisanje i odabir vođe grupe. Problem, koje je takođe već opisan, odnosi se na povećano opterećenje ovakvih čvorova. Sa druge strane, i u tehnikama rutiranja se dosta radi na izbegavanju čvorova koji su više opterećeni od drugih, što često nije moguće potpuno izbeći. Ukoliko bi se algoritmi učenja i rutiranja zajedno razvijali, postojao bi širi prostor za optimizaciju i raspodelu ukupnog opterećenja čvorova.

Naravno, ne treba zanemariti kompleksnost problema koji nastaje ukoliko se istraživanja i razvoj rade integralno (što je bio osnovni razlog zbog kojega je uvedena podela po slojevima). Prema tome, put ka prevazilaženju slojevite strukture sistema senzorskih mreža se mora prelaziti veoma postepeno. Slična situacija je očigledna i u svetu računarstva visokih performansi [Flynn, 2013]. Decenijama je rađeno na odvajanju dizajna hardvera i razvoja softvera. Međutim, ovakav pristup je dostigao svoje limite i postoji sve prisutniji konsenzus da će u budućnosti dominirati ko-dizajn hardvera i softvera.

Konačno, postoji potreba za razvojem matematičkih modela i slobodno dostupnih skupova podataka koji bi se mogli koristiti specifično za evaluaciju algoritama mašinskog učenja za bežične senzorske mreže. Prednosti ovakve infrastrukture bi bile značajne za više kategorija istraživača i korisnika vezanih za oblast mašinskog učenja i bežičnih senzorskih mreža. Oni koji su zainteresovani za primenu algoritma mašinskog učenja u bežičnim senzorskim mrežama bi imali koristi pošto bi postojali jasni načini za poređenje raznih

algoritama i pristupa. Oni koji rade na razvoju novih ideja i pristupa vezanih za distribuirano mašinsko učenje u bežičnim senzorskim mrežama bi dobili jasan standard za evaluaciju. Kao i mnoge druge grane naučnog istraživanja, razvoj u mašinskom učenju se oslanja na iterativni proces po principu pokušaja i greške. Ideja se razvije i potom evaluira. Rezultati evaluacije se zatim analiziraju kako bi se identifikovale eventualne slabosti i njihovi uzroci, kao i mogućnosti da se te slabosti otklone. Proces je dodatno otežan ako je u sklopu istraživanja neophodno razviti i testno okruženje i metrike performansi.

Reference:

- AKKAYA, K. and YOUNIS, M. 2005. A survey on routing protocols for wireless sensor networks *Ad hoc networks* 3, 3, 325-349
- BAHREPOUR, M., MERATNIA, N., POEL, M., TAGHIKHAZI, Z., and HAVINGA, P. J. M. 2010. Distributed event detection in wireless sensor networks for disaster management In: *Intelligent Networking and Collaborative Systems (INCOS)*, 2010 2nd International Conference on, 507-512
- BANERJEE, T., CHOWDHURY, K. R., and AGRAWAL, D. P. 2007. Using polynomial regression for data representation in wireless sensor networks: Research Articles *Int. J. Commun. Syst.* 7, 20, 829-856
- BERRY, C.C. 1992. The kappa statistic. *Journal of the American Medical Association*, 268, 18, 1992.
- BIOCCHI, N., MAMEI, N., AND F. ZAMBONELLI, F. 2007 Towards Self-organizing Virtual Macro Sensors, *First International Conference on Self-Adaptive and Self-Organizing Systems*, Los Alamitos, CA, USA 355--358, 2007
- BIOCCHI, N., MAMEI, N., AND F. ZAMBONELLI, F. 2007 Self-Organizing Spatial Regions for Sensor Network Infrastructures, *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops - Volume 02*, Washington, DC, USA 66--71, 2007
- BOUKERCHE, A. and SAMARAH, S. 2007. An Efficient Data Extraction Mechanism for Mining Association Rules from Wireless Sensor Networks In: *Communications, 2007. ICC'07. IEEE International Conference on*, 3936-3941
- CAMILO, T., SILVA, J., RODRIGUES A., BOAVIDA F. 2007. GENSEN: A Topology Generator For Real Wireless Sensor Networks Deployment. *5th IFIP Workshop on Software Technologies for Future Embedded & Ubiquitous Systems (SEUS 2007)*, pp. 436-445, Santorini, Greece.
- CATTERALL, E., VAN LAERHOVEN, K., AND STROHBACH, M. 2003. Self-organization in ad hoc sensor networks: an empirical study, *Proceedings of the eighth international conference on Artificial life*, Cambridge, MA, USA 260--263
- CHANDA, P., SUCHESTON, L., ZHANG, A., BRAZEAU, D., FREUDENHEIM, J., AMBROSONE, C., RAMANTHAN, M., 2008. AMBIENCE: A Novel Approach and Efficient Algorithm for Identifying Informative Genetic and Environmental Associations With Complex Phenotypes, *Genetics*, 180, 2, 1191-1210
- CHEN, S., HOVDE, D., KRISTEN A. PETERSON, K., ANDRÉ W. MARSHALL A., 2007. Fire detection using smoke and gas sensors. *Fire Safety Journal*, Volume 42, Issue 8, pp. 507-515

- CHEN, W. P., HOU, J. C., and SHA, L. 2004. Dynamic clustering for acoustic target tracking in wireless sensor networks *Mobile Computing, IEEE Transactions on* 3, 3, 258-271
- CHONG, S. K., GABER, M. M., KRISHNASWAMY, S., and LOKE, S. W. 2008. A rule learning approach to energy efficient clustering in wireless sensor networks In: *Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on*, 329-334
- DASGUPTA, K., KALPAKIS, K., and NAMJOSHI, P. 2003. An efficient clustering-based heuristic for data gathering and aggregation in sensor networks In: *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 1948-1953
- DE MEULENAER, G., GOSSET, F., STANDAERT, F., PEREIRA, O., 2008. On the Energy Cost of Communication and Cryptography in Wireless Sensor Networks, *WIMOB '08 IEEE International Conference on Wireless and Mobile Computing Networking and Communications*, October 12-14, 2008, pp. 580 - 585, Avignon, France.
- DI, M. and JOO, E. M. 2007. A survey of machine learning in Wireless Sensor networks From networking and application perspectives In: *6th International Conference on Information, Communications & Signal Processing, 2007*, 1-5
- FLOURI, K., BEFERULLOZANO, B., and TSAKALIDES, P. 2009. Optimal gossip algorithm for distributed consensus SVM training in wireless sensor networks In: *Proceedings of the 16th international conference on Digital Signal Processing IEEE Press, Piscataway, NJ, USA*, 886-891
- FLOURI, K., BEFERULLOZANO, B., and TSAKALIDES, P. 2006. Training a support-vector machine-based classifier in distributed sensor networks. In: *Proc. of 14th European Signal Processing Conf.*
- FLYNN, M., MENCER, O., MILUTINOVIC, V., RAKOCEVIC, G., STENSTROM, P., TROBEC, R., VALERO, M. Moving from petaflops (on simple benchmarks) to petadata per unit of time and power (on sophisticated benchmarks), accepted for publication, *Communications of the ACM*, 2013
- FRANK, A., ASUNCION, A., 2010 UCI Machine Learning Repository, Retrieved December 21, 2010, from UCI Machine Learning Repository: <http://archive.ics.uci.edu/ml>
- JAKULIN, A., BRATKO, I., 2004. Testing the Significance of Attribute Interactions. *Proceedings of the Twenty-first International Conference on Machine Learning (ICML-2004)*, pp. 409-416, Banff, Canada: ACM Press.
- JAKULIN, A., BRATKO I. 2003. Analyzing attribute dependencies. *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003)*, pp. 229-240, Dubrovnik, Croatia: Springer.
- GABER, M. M., ZASLAVSKY, A., and KRISHNASWAMY, S. 2005. Mining data streams: a review *ACM Sigmod Record* 2, 34, 18-26

- GAO, Q. BLOW, K., and HOLDING, D. 2004. Simple algorithm for improving time synchronisation in wireless sensor networks *Electronics Letters* 14, 40, 889-891
- GOEL, S. and IMIELINSKI, T. 2001. Prediction-based monitoring in sensor networks: taking lessons from MPEG *SIGCOMM Comput. Commun. Rev.* 5, 31, 82-98
- GRUENWALD, L., CHOK, H., and ABOUKHAMIS, M. 2007. Using Data Mining to Estimate Missing Sensor Data In: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops IEEE Computer Society, Washington, DC, USA, 207-212
- GUESTIN, C., BODI, P., THIBAU, R., PASKI, M., and MADDE, S. 2004. Distributed regression: an efficient framework for modeling sensor network data In: IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks ACM Press, 1-10
- HALATCHEV, M. and GRUENWALD, L. 2005. Estimating Missing Values in Related Sensor Data Streams In: COMAD'05, 83-94
- HONEINE, P., ESSOLOH, M., RICHARD, C., and SNOUSSI, H. 2008. Distributed regression in sensor networks with a reduced-order kernel model In: Proc. 51st IEEE GLOBECOM Global Communications Conference, New Orleans, LA, USA,
- JOHNSON, E. L. and KARGUPTA, H. 2000. Collective, Hierarchical Clustering from Distributed, Heterogeneous Data In: Revised Papers from Large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems, SIGKDD Springer-Verlag, London, UK, UK, 221-244
- KARGUPTA, H., DATTA, S., and GIANELLA, C. 2006. K-Means Clustering Over a Large, Dynamic Network In: Proceedings of 2006 SIAM Conference on Data Mining, Bethesda, MD,
- KOKIOPOULOU, E. and FROSSARD, P. 2011. Distributed Classification of Multiple Observation Sets by Consensus *Trans. Sig. Proc.* 1, 59, 104-114
- KUMARI, S. and SINGH, V. 2011. A Comprehensive Survey Paper on Sensor Data Mining Based on Sensor Network *International Journal of Technology* 1, 1, 37-41
- LI, Y., CHEN, C. S., SONG, Y.-Q., and WANG, Z. 2007. {Real-time QoS support in wireless sensor networks: a survey} In: 7th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems - FeT'2007, Toulouse, France
- LIM, T., LOH, W., SHIH, Y., 2000. A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms, *Machine Learning*, Volume 40, Number 3, pp. 203-228

- MCCONNELL, S., DATTA, S. 2005. A Distributed Approach for Prediction in Sensor Networks. *Workshop on Data Mining in Sensor Networks*, Newport Beach (CA), USA, 2005.
- MCCONNELL, S., DATTA, S. 2004. Building Predictors from Vertically Distributed Data. *Proceedings of the 14th Annual IBM Centers for Advanced Studies Conference*, pp. 150-162. Markham, Canada: IBM.
- Milutinovic, V. (1996) The best method for presentation of research results," *IEEE TCCA Newsletter*, pp. 1-6.
- MOSES, R. L., KRISHNAMURTHY, D., and PATTERSON, R. M. 2003. A self-localization method for wireless sensor networks *EURASIP Journal on Applied Signal Processing*, 348-358
- MUNAGA, H., MURTHY, J., and VENKATESWARLU, N. 2011. A Fault Tolerant Trajectory Clustering (FTTC) for selecting cluster heads in Wireless Sensor Networks *Arxiv preprint arXiv:1108.0747*,
- OTTO, C., MILENKOVIC, A., SANDERS, C., and JOVANOVIĆ, E. 2006. System architecture of a wireless body area network for ubiquitous health monitoring *Journal of Mobile Multimedia* 4, 1, 307-326
- PAPADIMITRIOU, S., SUN, J., and FALOUTSOS, C. 2005. Streaming pattern discovery in multiple time-series In: *Proceedings of the 31st international conference on Very large data bases VLDB Endowment*, 697-708
- PARK, B. H. and KARAGUPTA, H. 2005. Distributed Data Mining: Algorithms, Systems, and Applications In: *The Datamining Handbook*, 341-358
- PEDERSEN, R. and JUL, E. 2005. First International Workshop on Data Mining in Sensor Networks In: *2005 SIAM Conference on Data Mining*,
- RAKOCEVIC, G., ANDRIC, J., and MILUTINOVIC, V. 2011. Vertically Distributed Data Mining Approach to Event Detection in Wireless Sensor Networks. In: *X International Conference ETAI2011*, Ohrid
- VAN RIJSBERGEN, C. J. 1979. *Information Retrieval* (2nd ed.). Butterworth.
- ROEMER, K. 2007. Distributed mining of spatio-temporal event patterns in sensor networks In: *GIITG Workshop on Sensor Networks*,
- SAMARAH, S., BOUKERCHE, A., and REN, Y. 2008. Coverage-based Sensor Association Rules for Wireless Vehicular Ad hoc and Sensor Networks In: *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008*. IEEE
- SCHOLKOPF, B. and SMOLA, A. J. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* In: MIT Press, Cambridge, MA, USA,
- SKILLICORN, D. B. and MCCONNELL, S. M. 2008. Distributed prediction from vertically partitioned data *J. Parallel Distrib. Comput.* 1, 68, 16-36T
- SOUMAKAS, G. and VLAHAVAS, I. 2009. *Distributed Data Mining Database Technologies: Concepts, Methodologies, Tools, and Applications*, 157-171

- STANKOVIC, S., RAKOCEVIC, G., KOJIC, N., MILIĆEV, D., and VITAS, D., 2012 A Classification and Comparison of Data Mining Algorithms for Wireless Sensor Networks. *2012 IEEE International Conference on Industrial Technology (ICIT2012)*, 19-21.03.2012, Athens, Greece
- WU, S. and CLEMENTS-CROOME, D. 2007. Understanding the indoor environment through mining sensory data-A case study *Energy and Buildings* 11, 39, 1183-1191
- YUN, S. *et. al.* 2007 Study on the Applicability of Self-Organizing Maps to Sensor Network," *Proceedings of the 2007 ISIS Symposium*, May, 2007
- YOUNIS, O., KRUNZ, M., and RAMASUBRAMANIAN, S. 2006. Node clustering in wireless sensor networks: recent developments and deployment challenges *Network, IEEE* 3, 20, 20-25
- ZHANG, Y. and LEE, W. 2000. Intrusion detection in wireless ad-hoc networks In: *Proceedings of the 6th annual international conference on Mobile computing and networking*, 275-283
- ZHANG, Y., MERATNIA, N., and HAVINGA, P. 2010. Outlier detection techniques for wireless sensor networks: A survey *Communications Surveys & Tutorials, IEEE* 2, 12, 159-170

Biografija autora

Goran Rakočević je rođen 30. 05. 1983. godine u Beogradu (Srbija). Diplomirao je 2007. godine na Elektrotehničkom fakultetu Univerziteta u Beogradu, u oblasti Računarske tehnike i informatike. Od 2008. godine upisan je na doktorske studije na Elektrotehničkom fakultetu Univerziteta u Beogradu, na modulu za Računarsku tehniku i informatiku.

Od 2011. do 2013. godine je bio zaposlen na Matematičkom institutu Srpske akademije nauka i umetnosti, najpre u zvanju istraživača-pripravnika, a kasnije kao istraživač-saradnik. Angažovan je na projektu III44006: *Razvoj novih informaciono-komunikacionih tehnologija, korišćenjem naprednih matematičkih metoda, sa primenama u medicini, energetici, e-upravi, telekomunikacijama i zaštiti nacionalne baštine*, kao rukovodilac potprojekta *Razvoj inteligentih sistema baziranih na Data Mining strategijama*.

Počevši od 2007. godine Goran Rakočević je stekao značajno iskustvo u radu na međunarodnim projektima. Od 2011. godine angažovan je na Univerzitetu u Beogradu, u okviru projekta *BALCON: Boosting EU-Western Balkan Countries research collaboration in the Monitoring and Control area*, GA: 288076, finansiranom od strane Evropske komisije kroz Sedmi okvirni program. U periodu 2009–2011. godine bio je angažovan na Univerzitetu u Kragujevcu, u okviru projekta *ARTREAT: Multi-level patient-specific artery and atherogenesis model for outcome prediction, decision support treatment, and virtual hand-on training*, GA: 224297, takođe finansiranom od strane Evropske komisije kroz Sedmi okvirni program. U periodu 2007–2010. godine bio je angažovan na Elektrotehničkom fakultetu Univerziteta u Beogradu, u okviru projekta *ProSense: Promote, Mobilize, Reinforce and Integrate Wireless Sensor Networking Research and Researchers: Towards Pervasive Networking of West Balkan Countries and the EU*, GA: 205494, takođe finansiranom od strane Evropske komisije kroz Sedmi okvirni program.

Izjava o autorstvu

Potpisani Goran Rakočević
broj upisa 5001/08

Izjavljujem

da je doktorska disertacija pod naslovom

Mašinsko učenje u bežičnim senzorskim mrežama

- rezultat sopstevnog istraživačkog rada
- da predložena disertacija u celini ni u delovima nije bila predložena za dobijanje bilo koje diplome prema programima drugih visokoškolskih ustanova
- da su rezultati korektno navedeni i
- da nisam kršio autorska prava i koristio intelektualnu svojinu drugih lica

U Beogradu, 17.09.2014

Potpis doktoranda

Goran Rakočević

Izjava o istovetnosti štampane i elektronske verzije doktorskog rada

Ime i prezime autora	Goran Rakočević
Broj upisa	5001/08
Strudijski program	Računarska Tehnika i Informatika
Naslov rada	Mašinsko učenje u bežičnim senzorskim mrežama
Mentor	Prof. dr Veljko Milutinović

Potpisani Goran Rakočević

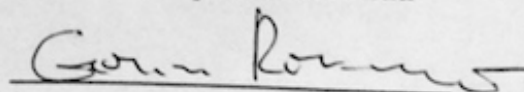
izjavljujem da je štampana verzija mog doktorskog rada istovetna elektronskoj verziji koju sam predao za objavljivanje na portalu **Digitalnog repozitorijuma Univerziteta u Beogradu**.

Dozvoljavam da se objave moji lični podaci vezani za dobijanje akademskog zvanja doktora nauka, kao što su ime i prezime, godina i mesto rođenja i datum odbrane rada.

Ovi lični podaci mogu se objaviti na mrežnim stranicama digitalne biblioteke, u elektronskom katalogu i u publikacijama Univerziteta u Beogradu.

U Beogradu, 17.09.2014

Potpis doktoranda



Izjava o korišćenju

Ovalšćujem Univerzitetsku biblioteku "Svetozar Marković" da u Digitalni repozitorijum Univerziteta u Beogradu unese moju doktorsku disertaciju pod naslovom:

Mašinsko učenje u bežičnim senzorskim mrežama
koja je moje autorsko delo.

Disertaciju sa svim priložima predao sam u elektronskom formatu pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju pohranjenu u Digitalni repozitorijum Univerziteta u Beogradu svi koji poštuju odredbe u odabranom tipu licence Kreativne zajednice (Creative Commons) za koju sam se odlučio:

1. Autorstvo
2. Autorstvo - nekomercijalno
3. Autorstvo - nekomercijalno - bez prerade
4. Autorstvo - nekomercijalno - deliti pod istim uslovima
5. Autorstvo - bez prerade
6. Autorstvo - deliti pod istim uslovima

U Beogradu, 17.09.2014

Potpis doktoranda

Geran Roman