



КЛАСИФИКАЦИЈА ТИПА ГРЕШКЕ СЕНЗОРА КОРИШТЕЊЕМ МЕТОДА МАШИНСКОГ УЧЕЊА

SENSOR FAULT CLASSIFICATION USING MACHINE LEARNING METHODS

Борис Јањић, Факултет техничких наука, Нови Сад

Област – РАЧУНАРСТВО И АУТОМАТИКА

Кратак садржај – У овом раду описани су сензори температуре и најчешће грешке које настају на сензорима. Такође су описане три методе машинског учења које су кориштене при класификацији типа грешке сензора. Кориштене методе су: k најближих сусједа, метод потпорних вектора и неуронска мрежа. Приказана је анализа поменутих метода и дато поређење добијених резултата.

Кључне речи: Грешке сензора, машинско учење, класификација, k најближих сусједа, метод потпорних вектора, неуронске мреже

Abstract – This paper describes temperature sensors and the most common faults on sensors. Three machine learning methods for classification are also described. Methods that has been used are: K Nearest Neighbors, Support Vector Machines and Neural Networks. The analysis of the mentioned classification methods of the sensor fault is presented and a comparison of the obtained results is given.

Keywords: Sensor fault, Classification, Support Vector Machines, K Nearest Neighbors, Neural Networks, Machine Learning

1. УВОД

Мјерење је настало као потреба да са природне појаве опишу квалитативним карактеристикама и резултат је опажања и поређења тих појава. Појавом управљања и аутоматизације процеса јавља се потреба да се мјерења величина која нису електрична некако пребаце у електрични облик. Тај проблем се рјешава уз помоћ сензора. Нагли развој сензора почиње у периоду аутоматизације и у компјутерској ери. Данас постоји широк дијапазон сензора од најпростијих давача и биметала до крајње софистицираних сензора који се користе за најпрецизнија мјерења. Због велике употребе сензора велика штета може да настане ако се благовремено не детектује квар на сензору. У овом раду биће ријечи о најчешћим типовима грешака које се могу десити на сензорима.

Такође ће се говорити и о машинском учењу и класификационим методама. Биће ријечи о прикупљању, припреми и обради података са грешкама сензора. Биће приказане анализе претходно поменутих метода машинског учења и резултати њиховом примјеном на улазне податке.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Велимир Чонградац, ред. проф.

2. СЕНЗОРИ

Сензор је уређај који претвара мјерену физичку величину у електрични сигнал, одликује се малим димензијама, изузетним техничким карактеристикама и способношћу обраде сигнала [1].

Значај сензора се огледа у прикупљању података о објекту или процесу зарад одређених анализа, дијагностике, пројектовања, управљања. Примјена им је веома широка у свим инжењерским и многим другим областима.

2.1. Сензори температуре

Температура је једна од основних седам физичких величина SI (међународни систем јединица) система. Сензори температуре према изведби могу бити: термоелементи, експанзиони, отпорнички, полупроводнички, сензори инфрацрвеног зрачења.

2.2. Грешке сензора

Одступања од очекиваног понашања на излазу сензора се сматрају грешкама. Због утицаја многобројних фактора, чије се дејство не може контролисати, услови у којима раде системи се непрекидно мијењају. Зато резултат мјерења увијек садржи грешку.

Аутори у радовима [2], [3] и [4] наводе сљедеће грешке сензора као најчешће: грешка дрифта, грешка биаса, грешка деградације прецизности, грешка пика и грешка заглављења.

Нека је $s(t) = h(t) + \eta$ очекивано понашање сензора без присуства грешке, гдје је $h(t)$ излазна вриједност сензора у времену t , и $\eta \sim N(0, \delta_\eta^2)$ је шум. На основу ове једначине моделоваћемо једначине горе наведених грешака.

Дрифт је промјена статичке карактеристике сензора у условима употребе у току дужег временског периода

$$S(t) = h(t) + \eta + b(t), b(t) = b(t-1) + \vartheta, \vartheta = const \quad (1)$$

Грешка биаса (eng. *Bias Fault*) се моделује тако што се на излаз сензора дода нека константа. Ова грешка се јавља због хардверских проблема као што су кратак спој или грешка калибрисања. Моделује се:

$$S(t) = h(t) + \eta + \vartheta, \vartheta = const \quad (2)$$

Грешка деградације прецизности (eng. *Precision Degradation Fault*) се јавља усљед високо фреквентних шумава или физичког оштећења сензора. Математички модел:

$$S(t) = h(t) + \eta + \vartheta \vartheta \sim N(0, \delta_\vartheta^2) \delta_\vartheta^2 \gg \delta_\eta^2 \quad (3)$$

δ_n^2 представља варијансу високо фреквентног шума и показује да она мора бити вишеструко већа од δ_n^2 варијансе шума сензора који смо у стању да коригујемо.

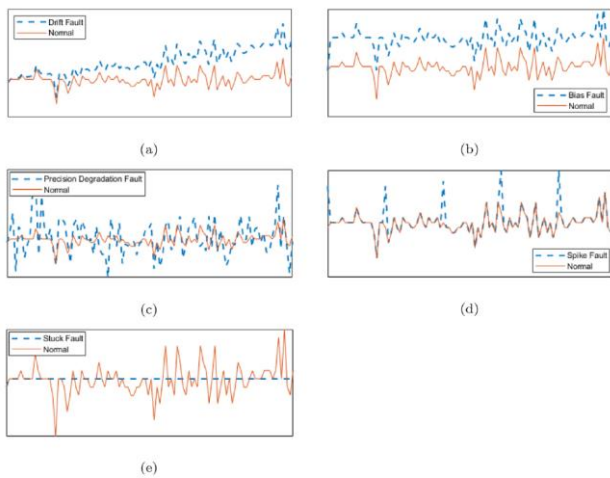
Грешка пика (eng. *Spike Fault*) је грешка гдје се на излазу сензора, у константним интервалима, појављују пикови велике амплитуде. Математички је описујемо:

$$S(t) = \begin{cases} h(t) + \eta + \vartheta(t); & \forall t = v \times \tau, v = \{1, 2, \dots\}, \tau \geq 2, \\ h(t) + \eta; & \text{otherwise} \end{cases} \quad (4)$$

Заглављење (eng. *Stuck Fault*) како само име каже манифестује се тако што на излазу сензора даје константну вриједност. Може да буде привремена или трајна. Моделује се на сљедећи начин:

$$S(t) = \vartheta, \vartheta = \text{const} \quad (5)$$

На слици 1 је приказано свих 5 грешака сензора које су описане. Поменуте грешке ће бити кориштене у експерименталном дијелу задатка.



Слика 1. Типови грешке сензора: (а) дрифт, (б) биас, (ц) деград. прецизности, (д) пик, (е) заглављење [3]

3. МАШИНСКО УЧЕЊЕ

Машинско учење је област вјештачке интелигенције која се бави изградњом рачунарских система који уче из искуства без експлицитног програмирања.

За компјутерски програм се каже да учи из искуства E (*Experience*), везаног за задатак T (*Task*), и мјеру перформанси P (*Performance*), уколико се његове перформансе на задатку T , мјерене метрикама P , унапређују са искуством E [5].

Алгоритме машинског учења дијелимо на: надгледано учење, ненадгледано учење и учење условљавањем.

Надгледано учење је облик машинског учења који као скуп параметара за тренирање прима улазне податке и жељене очекиване вриједности на излазу. Циљ је добити резултате за необиљежене податке [5].

3.1. Алгоритми класификације

Класификација представља једну од двије основне врсте надгледаног учења.

Класификација је проблем предвиђања категоричке циљне промјенљиве. Категоричким се сматрају промјенљиве које узимају коначан број вриједности међу којима нема уређења [5]. Конкретно у овом раду

је ријеч о класификовању типова грешке сензора. Скуп грешака је дат у поглављу о грешкама, а резултати ће бити изнесени у поглављу везаном за имплементацију.

Када излаз модела може имати више од двије класе онда говоримо о *multiclass* класификацији.

У наставку ће бити више ријечи о алгоритмима који су кориштени у практичном дијелу рада: k најближих сусједа, методу потпорних вектора и неуронским мрежама.

3.1.1. K најближих сусједа

K најближих сусједа (eng. *K Nearest Neighbors* - KNN) је вјероватно најједноставнији алгоритам машинског учења који се може користити и за класификацију и за регресију. Основна претпоставка овог алгоритма је постојање растојања над простором атрибута. Најчешће се претпоставља векторска репрезентација инстанци и еуклидско растојање, али могуће су и општије претпоставке [6].

3.1.2. Метод потпорних вектора

Метод потпорних вектора (eng. *Support Vector Machine* - SVM) је један од важнијих метода машинског учења. SVM је иницијално креиран као класификациони проблем двије класе. Ова метода се проширује на проблем више класа (eng. *multiclass*), а самим тим постаје све занимљивија тема истраживања [7].

Заснована на геометријској интуицији, ова метода тражи најбољу раздвајајућу хиперраван. [6].

3.1.3. Неуронске мреже

Неуронске мреже (*neural networks* - NN) се убрајају међу најпопуларније и најпримјењеније методе машинског учења. Њихове примјене су многобројне и помјерају домете вјештачке интелигенције, рачунарства и примјењене математике. Неке од њихових примјена су класификација текста и слика, аутономна возња, препознавање говора, и слично. Неуронске мреже заправо представљају параметризовану репрезентацију која може да послужи за апроксимацију других функција. Проналажење одговарајућих параметара врши се математичком оптимизацијом неког критеријума квалитета апроксимације [6].

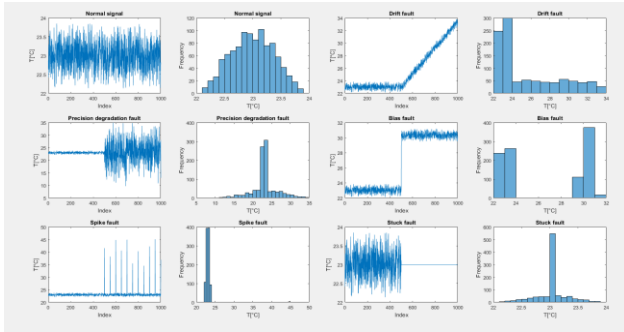
4. КЛАСИФИКАЦИЈА ГРЕШКЕ СЕНЗОРА

Идеја овог задатка јесте да се симулирају најчешћи типови грешака на сензорима и да се класификују. Циљ је да се подаци обраде и прикажу анализе методама машинског учења. Конкретно ће бити обрађене методе KNN, SVM и NN и биће приказани резултати.

4.1. Прикупљање и припрема података

Процес прикупљања података и припреме за класификационе методе је веома сличан приступу у радовима [2], [3] и [4]. Мјерења су изведена користећи *Arduino Uno* и сензор температуре и влажности DHT22, с тим да је акценат био на мјерењу температуре. Мјерења су прикупљена користећи серијску комуникацију између рачунара и *Arduino*

Uno микроконтролера. Извршено је 100 мјерења по 1000 тачака на сензору који нема никакве грешке у раду. Мјерења су била прилично чиста, пошто у просторији гдје је вршено мјерење није било неке веће промјене температуре. Ради симулирања шума у реалним системима, додат је шум у нормалан сигнал. Наредни корак је био да се додају грешке у нормалан сигнал. Додате су грешке описане једначинама (1), (2), (3), (4) и (5) и примјери сваког сигнала су дати на слици 14. Као што се види на сликама, у првих 500 тачака сигнал је нормалан сигнал и није модификован, док је у других 500 тачака додата грешка.



Слика 2. Прикази грешака и њихових хистограма

Послије проширења скупа са уметнутим грешкама, гдје је сваки тип грешке представљен једном класом читав скуп података износи 600x1000. Пошто је скуп од 1000 тачака прилично велик, на основу тих тачака креирано је 12 карактеристика које ће бити кориштене као улазни параметри у класификаторе, па су димензије сета података редуковане на 600x12. једначине које су кориштене су приказане у табели 1, а средње вриједности карактеристика су приказане у табели 2.

Табела 1. Списак једначина кориштених у одређивању карактеристика

Назив	Дефиниција	Назив	Дефиниција
Средња вриједност	$\mu = \frac{1}{N} \sum_{i=1}^N x_i$	Стандардна девијација	$Y_{STD} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$
Средња квадратна вриједност	$Y_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$	Варијанса	$Y_{VAR} = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$
Кв. коријен амплитуде	$Y_{SRA} = \left(\frac{1}{N} \sum_{i=1}^N \sqrt{ x_i } \right)^2$	Оштрина врха	$Y_{KV} = \frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - \mu}{Y_{STD}} \right)^4$
Накривљеност	$Y_{SV} = \frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - \mu}{Y_{STD}} \right)^3$	Фактор гребена	$Y_{CF} = \max \left(\frac{ x_i }{Y_{RMS}} \right)$
Фактор импулса	$Y_{IF} = \max \left(\frac{ x_i }{\frac{1}{N} \sum_{i=1}^N x_i } \right)$	Гранични фактор	$Y_{MF} = \max \left(\frac{ x_i }{Y_{SRA}} \right)$
Фактор форме	$Y_{FPV} = \frac{Y_{RMS}}{\mu}$	Од пика до пика	$Y_{PPV} = \max(x_i) - \min(x_i)$

4.2. Методологија евалуације

Модел је увијек потребно процијенити на новим и непознатим подацима. У фази обуке моделу дајемо улазне податке и очекивану вриједност излаза и на основу тога се модел обучава. У фази евалуације модела користи се други сет података, а као улаз у модел користе се само улазни подаци. На основу тих података модел одређује излаз система. Добијене вриједности се пореде са очекиваним и добија се

тачност модела. У овом раду кориштене су следеће метрике перформанси: прецизност (*Precision*), одзив (*Recall*), Ф1 скор (*F1 score*) и тачност (*Accuracy*).

Табела 2. Средње вриједности карактеристика за сваки тип грешке

Карактеристика	Нормални	Дрифт	Деградиација прецизности	Бис	Пик	Заглављење
μ	22.9994	29.0187	23.0059	26.6089	23.1924	22.9998
Y_{STD}	0.4030	7.7988	3.5599	3.6331	1.9266	0.2850
Y_{RMS}	23.0029	30.2908	23.2796	26.8875	23.2725	23.0015
Y_{VAR}	0.1625	85.0076	12.6816	15.5133	3.7278	0.0813
Y_{SRA}	22.9976	28.4289	22.8627	26.4674	23.1627	22.9989
Y_{KV}	2.5210	2.4035	5.0826	1.1041	98.0002	5.0326
Y_{SV}	-0.0003	0.8623	0.0065	-0.0036	9.4265	-0.0038
Y_{CF}	1.0460	1.5278	1.5495	1.1590	2.0129	1.0447
Y_{IF}	1.0461	1.5964	1.5679	1.1710	2.0199	1.0448
Y_{MF}	1.0462	1.6314	1.5778	1.1772	2.0225	1.0448
Y_{FF}	1.0002	1.0397	1.0119	1.0101	1.0035	1.0001
Y_{PPV}	2.1263	25.5666	26.0597	9.2622	24.9120	2.0593

4.3. Анализа експеримента

За имплементацију класификатора кориштен је програмски језик *python 3.7* верзија и кориштена је библиотека *scikit-learn*. Циљ је да се упореде резултати класификације следећих метода KNN, SVM и NN. Ради вјеродостојности резултата за сваку метода користи се исти скуп за тренирање и исти скуп за тестирање. Извршена су три експеримента над подацима:

- Први експеримент је био над чистим подацима, гдје су улази у сваки од класификатора били сирови подаци. Сваки класификатор имао је 1000 улазних параметара.
- Други експеримент је био извлачење најбољих параметара од 1000 задатих. За одабир 10 и 50 најбољих представника од 1000 је кориштен *SelectKBest* селектор. Улаз у класификаторе је био величине 10 и 50 карактеристика.
- За трећи експеримент су употријебљене претходно припремљене карактеристике. Једначине карактеристика су дате у табели 2. У овом случају је узиман варијабилни број карактеристика од 5 до 12.

Скалирање података доприноси томе да нека од карактеристика не буде значајно доминантнија од друге па су подаци прескалирани користећи *MaxAbsScaler*. Од укупног броја података (600) приказаћемо анализе гдје је 20%, 40% и 60% укупног броја података одвојено за тестирање модела.

4.3.1. К најближих сусједа

Имплементиран користећи *KNeighborsClassifier*. Метрике су рађене на основу параметра К и величине тестног скупа. Експериментално је утврђено да је најбољи избор за параметар К=9.

3.1.1. Метод потпорних вектора

За имплементацију ове методе кориштен је *SVC*. За потребе ових експеримената биће кориштени *linear*,

poly и *rbf* типови *kernela*. Пошто се ради о *multiclass* класификацији изабрана је и one-vs-one стратегија, која дијели *multiclass* проблем у један бинарни задатак класификације за сваки пар класа.

4.3.3. Неуронске мреже

За моделовање неуронске мреже кориштен је *MLPClassifier*. Употријебљени *solver* је *adam*, а као активациона функција је узета *relu*. Неуронска мреже је састављена из три слоја, улазног, излазног и скривеног слоја. Улазни слој неуронске мреже зависи од броја карактеристика, што ће у првом експерименту бити 1000, у другом 10, а у трећем редом 5, 8, 10 и 12. Излазни слој једнак је броју класа и у свим експериментима износи 6. Кориштен је предефинисани број неурона у скривеном слоју и он износи 100.

4.4. Резултати

Кренувши од првог експеримента, који је као улазне параметре узимао сирове податке, видимо да је тачност за KNN 52-55% у зависности од параметра K, за SVM је 55-80% у зависности који *kernel* је кориштен, а за NN класификатор је тачност 70-72% у зависности од величине скупа за обучавање. Видимо да сваки од класификатора са прецизношћу од готово 100% класификује грешке дрифта, пика и биаса. Битна ствар за први експеримент је и то што сваки класификатор има проблем да класификује три класе нормалан сигнал, грешку деградације прецизности и грешку заглављења. Једино је SVM *rbf* успио да ријеша проблем грешке деградације прецизности, док су друге двије класе остале неријешене.

Због осредње тачности и проблема класификације првенствено горе наведене три проблематичне класе, предложен је други експеримент гдје је узето 50 и 10 најбољих карактеристика из сировог сигнала, а остале су одбачене. Над KNN класификатором је показано како је класификатор са 10 тачака показао боље перформансе. Када упоредимо резултате KNN, SVM и NN класификатора можемо видјети да се тачност повећала.

Тачност KNN класификатора је скочила са 55% на 84%, а дјелимично је ријешен проблем класификовања проблематичних класа, док су за остале 3 класе перформансе остале исте. За SVM класификатор је слична ситуација као за KNN класификатор, перформансе су поправљене, посебно за *kernel poly* гдје је тачност скочила са 55% на 91-93%. Класификација по критичним класама се знатно поправила у односу на претходни експеримент. Код NN класификатора у овом експерименту није било пуно напретка, тачност је скочила са 72% на 78%, с тим да проблем класификовања нормалног сигнала и закочења није ријешен.

Пошто је у другом експерименту одбачен огроман број тачака, од 1000 тачака сигнала кориштено је само 10, тако је огроман потенцијал који имају остале тачке остао неискориштен. Због тога су кориштене математичке једначине из табеле 1 да се направи 12 карактеристика на основу сировог сигнала.

У трећем експерименту се показало да се са таквим карактеристикама, небитно да ли их узмемо само 5 или 12, добија тачност од 99-100%. Такође је битно напоменути да повећање обучавајућег скупа са 40% на 80% не утиче битно на перформансе добијених резултата.

5. ЗАКЉУЧАК

У првом експерименту је показано је да су метрике које се добију из чистих података задовољавајуће за неке класе грешка као што су дрифт, биас и пик, док за друге грешке то није случај. Показано је и да се перформансе могу поправити ако се из цијелог скупа података извуку најбољи представници, као што је урађено у другом експерименту. Тај приступ се показао добрим јер су перформансе на свим свим методама скочиле.

Најбоље се показао метод потпорних вектора са тачношћу од 93%. У трећем експерименту је показано да је могуће искористити све вриједности сигнала за израчунавање одређених карактеристика на основу којих ће се обучаваати класификатори, а као резултат тога перформансе свих класификатора износе 99-100%.

Потенцијална унапређења и даљи кораци би били да се користе прави подаци са реалним грешкама, и да се размотре перформансе постојећих модела. Још један приједлог за даљи развој био би да се смањи број карактеристика за обучавање модела са задржавањем истих или побољшањем перформанси.

4. ЛИТЕРАТУРА

- [1] М. Поповић „*Senzori u robotici*“, Виша електротехничка школа, Београд, 1996
- [2] S. U. Jan, I. Koo, „A Novel Feature Selection Scheme and a Diversified-Input SVM-Based Classifier for Sensor Fault Classification“
- [3] S. U. Jan, Y. D. Lee, I. S. Koo, „A distributed sensor-fault detection and diagnosis framework using machine learning“
- [4] S. U. Jan, Y.-D. Lee, J. Shin, and I. Koo, „Sensor fault classification based on support vector machine and statistical time-domain features“
- [5] B. Nikolić, D. Drašković „*Nastavni materijali iz predmeta Inteligentni sistemi*“, ETF, Београд 2021.
- [6] M. Nikolić, A. Zečević „*Mašinsko učenje*“, Matematički fakultet, Београд 2019.
- [7] Y. Zhang „*Support Vector Machine Classification Algorithm and Its Application*“

Кратка биографија:



Борис Јањић рођен је у Требињу 1994. године. Факултет техничких наука уписао је 2013. године. Дипломски рад из области Електротехника и рачунарство – Аутоматика и управљање системима одбранио је 2017. године.