ВЕШТАЧКА ИНТЕЛИГЕНЦИЈА
/
ARTIFICIAL INTELLIGENCE

(ВИ/VII)

Rešavanje problema ekonomične raspodele snaga generatora primenom fazorske optimizacije roja čestica

Milena Jevtić, Miroljub Jevtić, Jordan Radosavljević, Sanela Arsić i Dardan Klimenta

Apstrakt— **Minimizacija troškova goriva i emisije štetnih gasova u termoelektranama podešavanjem izlaznih snaga generatora je jedan od važnih problema u upravljanju elektroenergetskim sistemima. Ovaj problem je poznat kao Combined economic emision dispach (CEED) problem. U ovom radu je za rešavanje CEED problema predložen meta-heuristički algoritam pod nazivom Fazorska optimizacija roja čestica, koji predstavlja unapređenu varijantu Optimizacije roja čestica. Parametri Fazorske optimizacije roja čestica se tokom iteracija automatski podešavaju pa je ovaj algoritam, adaptivni i neparametarski, što je njegova prednost. Performanse predloženog algoritma za rešavanje CEED problema se u radu ocenjuju na standardnom IEEE test sistemu sa 30 čvorova i 6 generatora. Na osnovu dobijenih rezultata utvrđeno je da ovaj algoritam ima bolje karakteristike od algoritama koji su primenjeni u drugim publikovanim radovima za rešavanje CEED problema.**

Ključne reči **- Combined economic emision dispach; Fazorska optimizacija roja čestica; Upravljanje elektroenergetskim sistemima.**

I. Uvod

Ekonomična raspodela snaga generatora sa istovremenom minimzacijom emisije štetnih gasova (eng. Combined Economic and Emission Dispatch (CEED)) predstavlja podešavanje izlaznih snaga određenog broja generator u termoelektranama, pri zadatom opterećenju i pri zadatim ograničenjima u sistemu, minimizirajući troškove goriva i emisiju štetnih gasova. Funkcije koje opisuju emisiju štetnih gasova i troškove goriva, uzimajući u obzir efekat ventila u elektrani, su nelinearne i nekonveksne tako da je CEED problem u literaturi rešavan metaheurističkim optimizacionim algoritmima koji daju približno rešenje. U publikovanim radovima je predložen veći broj metaheurističkih algoritama u cilju dobijanja što tačnijeg i što bržeg rešenja ovog problema [1], [2], [3]. Brzina i tačnost ovih algoritama utiču na kvalitet softvera, u koje se inkorporiraju, a koji služe za upravljanje emisijom gasova i troškovima goriva u termoelektrani.

U ovom radu se za rešavanje CEED problema predlaže primena jednog od najnovijih algoritama, Fazorske optimizacije roja čestica (eng. Phasor particle swarm optimizacition (PPSO)) [4].

Cilj ovog rada je da se pokaže da PPSO može efikasno da se primeni za rešavanje CEED problema i da daje bolje rezultate u odnosu na druge algoritme koji su u literature primenjivani za rešavanje istog problema.

II. CEED model

Funkcija troškova goriva generatora u termoelektrani obično ima kvadratni oblik:

$$F_g\left(P_g\right) = a_g + b_g P_g + c_g P_g^2, \qquad g = 1, 2, \ldots, G \qquad (1)$$

gde su: F_g (\$/h) troškovi goriva g- tog generator, P_g (MW) izlazna snaga g-tog generatora, a_g, b_g i c_g koefficienti.

Funkcija $F_g\,(P_g)$ postaje ne-konveksna kada se uzme u obzir promena snage zbog sekventnog otvaranja ventila u termoelektrani (efekat ventila) [5]:

$$F_g\left(P_g\right) = a_g + b_g P_g + c_g P_g^2 + \left| d_g \sin\left(e_g\left(P_g^{\min} - P_g\right)\right)\right| \qquad (2)$$

gde su: d_g i e_g koeficijenti koji se odnose na efekat ventila i P $P_g^{\min}$ donja granična snaga g-tog generatora.

Funkcija koja modeluje emisiju gasova u termoelektrani se predstavlja kao zbir kvadratne i eksponencijalne funkcije izlazne snage generatora [6], [7]:

$$E_g\left(P_g\right) = \alpha_g + \beta_g P_g + \eta_g P_g^2 + \xi_g \exp\left(\lambda_g P_g\right) \qquad (3)$$

gde su: E_g (t/h) količina gasova koji se emituju tokom rada g-tog generatora, P_g (MW) izlazna snaga g-tog generatora, i α_g, β_g, η_g, ξ_g *i* λ_g emisioni koeficijenti.

Ako se (1) i (2) kombinuju sa (3), dobija se sledeća funkcija [8]:

Milena Jevtić – Tehnički fakultet u Boru, Univerzitet u Beogradu, Vojske Jugoslavije 12, 19210 Bor, Srbija (e-mail: mjevtic@tfbor.bg.ac.rs).

Miroljub Jevtić – Fakultet tehničkih nauka u Kosovskoj Mitrovici, Univerzitet u Prištini, Knjaza Miloša 7, 38220 Kosovska Mitrovica, Srbija (e-mail: miroljub.jevtic@pr.ac.rs).

Jordan Radosavljević - Fakultet tehničkih nauka u Kosovskoj Mitrovici, Univerzitet u Prištini, Knjaza Miloša 7, 38220 Kosovska Mitrovica, Srbija (e-mail: jordan.radosavljevic@pr.ac.rs).

Sanela Arsić – Tehnički fakultet u Boru, Univerzitet u Beogradu, Vojske Jugoslavije 12, 19210 Bor, Srbija (e-mail: saarsic@tfbor.bg.ac.rs).

Dardan Klimenta - Fakultet tehničkih nauka u Kosovskoj Mitrovici, Univerzitet u Prištini, Knjaza Miloša 7, 38220 Kosovska Mitrovica, Srbija (e-mail: dardan.klimenta@pr.ac.rs).

$$FE = w\sum\nolimits_{g\in G} F_g\left(P_g\right) + \left(1-w\right)\gamma\sum\nolimits_{g\in G} E_g\left(P_g\right) \tag{4}$$

gde su: γ factor skaliranja, *w* težinski faktor čija vrednost se uzima u granicama 0 < *w* < 1, i *G* je ukupni broj generatora koji se razmatraju, priključenih na system. CEED problem se rešava tako što se izabere faktor *w* a zatim minimizuje funkcija (4). Izborom gornje granice težinskog faktora, *w* = 1, minimizuje se samo funkcija F_g (P_g), izborom donje granice, *w* = 0, minimizuje se samo funkcija E_g (P_g), dok izbor drugih vrednosti težinskog faktora odgovara istovremenoj minimizaciji troškova goriva i emisije gasova. Faktor skaliranja *γ* je uveden da bi se funkcija (4), rešavala kao jednociljni optimizacioni problem umesto kao dvociljni.

Minimizacija se vrši za zadate granice snage svakog generator, tj.

$$P_g^{\min} \le P_g \le P_g^{\max} \tag{5}$$

gde su: $P_g^{\min}$, $P_g^{\max}$ i P_g su minimalna, maksimalna i stvarna snaga *g*-tog generatora, i za zadatu ravnotežu između proizvedene snage i utrošene snage, tj.

$$\sum\nolimits_{g\in G} P_g - P_D - P_{loss} = 0, \tag{6}$$

gde su: P_D ukupna snaga svih potrošača i P_{loss} gubici snage u prenosnom sistemu.

Gubici snage u prenosnom sistemu, P_{loss} , se izražavaju kao kvadratna funkcija trenutne snage generatora, tj. iz Kronove formule gubitaka [8], kao:

$$P_{loss} = \sum\nolimits_{g\in G}\sum\nolimits_{j\in G} P_g B_{gj} P_j + \sum\nolimits_{g\in G} B_{0g} P_g + B_{00} \tag{7}$$

gde su B_{gj} i B_{0g} koeficijenti *B*-loss matrice a B_{00} je konstanta.

Da bi se zadovoljilo ograničenje (6), tokom iterativnog procesa optimizacije, jedan od generatora (npr. generator *G*) je odabran kao zavisni (labav) generator. Za taj generator vrednost izlazne snage, P_G, se računa iz sledeće jednačine:

$$P_G = P_D + P_{loss} - \sum\nolimits_{g=1}^{G-1} P_g \tag{8}$$

Gubici snage, P_{loss}, se onda dobijaju na sledeći način: (i) zadavanje početne vrednosti $P_{loss} = P_{loss}^{(0)} = 0$ u (8), (ii) određivanje vrednosti $P_G^{(0)}$ iz (8) za $P_{loss} = P_{loss}^{(0)} = 0$, (iii) izračunavanje nove vrednosti $P_{loss}^{(1)}$ primenom (7), (iv) proveravanje da li je razlika između dve uzastopne vrednosti gubitaka snage manja ili jednaka zadatoj dozvoljenoj oleranciji *δ*, tj.

$$\left|P_{loss}^{(1)} - P_{loss}^{(0)}\right| \le \delta \tag{9}$$

i (v) izračunavanje vrednosti $P_G^{(1)}$ primenom (8) za $P_{loss} = P_{loss}^{(1)}$. Ako je razlika $\left|P_{loss}^{(1)} - P_{loss}^{(0)}\right|$ manja ili jednaka zadatoj toleranciji *δ*, ograničenje (6) koje predstavlja ravnotežu snaga, je zadovoljeno. U suprotnom, procedura se ponavlja. Kada je vrednost P_G izračunata, potrebno je proveriti da li se vrednost P_G nalazi u odgovarajućim granicama (5). Zatim se definiše promenljiva, $P_G^{\lim}$, na sledeći način:

$$P_G^{\lim} = \begin{cases} P_G^{\max} & ako\ je \quad P_G > P_G^{\max} \\ P_G^{\min} & ako\ je \quad P_G < P_G^{\min} \\ P_G & ako\ je \quad P_G^{\min} \le P_G \le P_G^{\max} \end{cases} \tag{10}$$

Da bi se osiguralo da zavisna promenljiva P_G ostaje u zadatim granicama, funkciji cilja (4) se dodaje kvadratni penalni član sa penalnim faktorom, λ_p. Na taj način se dobija proširena funkcija cilja:

$$FE_p = FE + \lambda_p\left(P_G - P_G^{\lim}\right)^2 \tag{11}$$

III. PSO I PPSO

Optimizacija roja čestica (eng. Particle swarm optimization (PSO)) je inspirisana ponašanjem rojeva u prirodi u potrazi za hranom [9]. Jedinka u jatu menja svoju poziciju i brzinu kretanja postepeno se krećući ka izvoru hrane. U PSO, svaka jedinka (čestica) u jatu je predstavljena vektorima pozicije i brzine, na sledeći način:

$$X_i(t) = \left[x_i^1(t),\ldots,x_i^k(t),\ldots,x_i^n(t)\right] \tag{12}$$

$$V_i(t) = \left[v_i^1(t),\ldots,v_i^k(t),\ldots,v_i^n(t)\right] \tag{13}$$

gde su $X_i\,(t)$ i $V_i\,(t)$ vektor pozicije i vektor brzine *i*-te čestice u vremenu (iteraciji) *t*; $x_i^k(t)$ i $v_i^k(t)$ su pozicija i brzina *i*-te čestice *k*-te dimenzije. Početne vrednosti vektora su slučajno odabrane. Brzine i pozicije čestica u narednoj iteraciji su određene pomoću sledećih jednačina:

$$\begin{aligned} v_i^k(t+1) &= w(t)v_i^k(t) + C_1 r_1\left(pbest_i^k(t) - x_i^k(t)\right) + \\ &+ C_2 r_2\left(gbest^k(t) - x_i^k(t)\right) \end{aligned} \tag{14}$$

$$x_i^k(t+1) = x_i^k(t) + v_i^k(t+1) \tag{15}$$

U (14), w (t) je inercijalna težina, C_1 i C_2 su parametri regulacije ubrzanja čestica, r_1 i r_2 su are the uniformno raspoređeni slučajni brojevi unutar granica [0,1], $pbest_i^k(t)$ je najbolja pozicija i-te čestice k-te dimenzije (indiviualna najbolja pozicija), i $gbest^k$ je globalno najbolja pozicija u celoj populaciji. Drugi član u (14) predstavlja exploracioni deo PSO. Inercijalna težina vrši uravnoteženje između lokalnog i globalnog pretraživanja rešenja. U početnom stadijumu procesa pretraživanja vrednost w je velika kako bi se pojačala globalna eksploracija. U poslednjem stadijumu vrednost w se smanjuje kako bi se dobila bolja lokalna exploracija.

Algoritam PPSO je predložio Gholamghasemi M. sa koautorima 2019. godine [4]. Parametri C_1 i C_2, koji se u algoritmu PSO zadaju ručno, u PPSO algoritmu su modelovani faznim uglom (θ) definisanim u teoriji fazora. Na taj način, PPSO, za razliku od PSO, postaje adaptivni i neparametarski algoritam. Vrednost w (t) je u PPSO jednaka nuli. Brzina u svakoj iteraciji se ažurira na sledeći način.

$$\begin{aligned}V_i(t)&=\left|\cos\theta_i(t)\right|^{2\cdot\sin\theta_i(t)}\times\left(Pbest_i(t)-X_i(t)\right)+\\&+\left|\sin\theta_i(t)\right|^{2\cdot\cos\theta_i(t)}\times\left(Gbest(t)-X_i(t)\right)\end{aligned}\qquad(16)$$

gde su: $Pbest_i(t)$ i $Gbest(t)$ vektori individualne i globalne najbolje pozicije; X_i (t) je vektor trenutne pozicije i-te čestice u t-toj iteraciji; θ_i jedno-dimenzioni fazni ugao vektora $\vec{X}_i\angle\theta_i$ za i-tu česticu. Za početnu populaciju koja se sastoji od N čestica (za t = 1), vektor $\vec{X}_i$ je: $\vec{X}_i=|X_i|\angle\theta_i$ (i = 1:N). Na početku pretraživanja rešenja, generisano je N slučajnih čestica (rešenja) u n-dimenzionom prostoru problema sa faznim uglom θ_i dobijenim iz ravnomerne raspodele $\theta_i = U$ (0, 2π), i sa početnom granicom brzine $V_{i,max}$. Donja i gornja granica V_i (t) su definisane sledećim intervalom [-$V_{i,max}$ (t), $V_{i,max}$ (t)].

Pozicije čestica se ažuriraju pomoću sledeće jednačine:

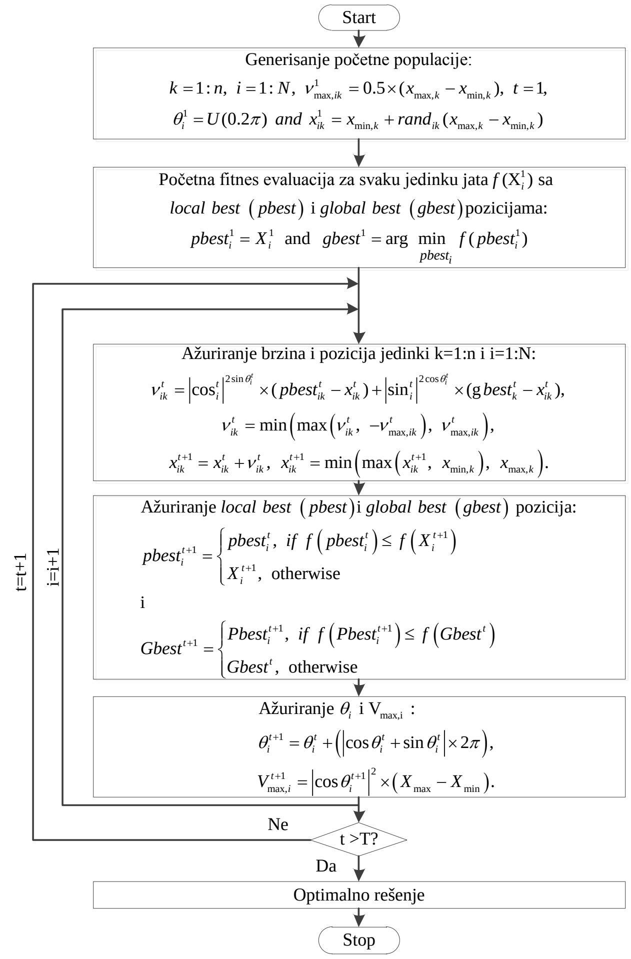
$$\vec{X}_i(t+1)=\vec{X}_i(t)+\vec{V}_i(t)\qquad(17)$$

Posle ažuriranja brzine čestice i pozicije primenom (16) i (17), fazni ugao θ_i i maksimalna brzina $V_{i,max}$ za sledeću iteraciju izračunavaju se iz sledećih jednačina:

$$\theta_i(t+1)=\theta_i(t)+\left|\cos\theta_i(t)+\sin\theta_i(t)\right|\times(2\pi)\qquad(18)$$

$$V_{i,\max}(t+1)=\left|\cos\theta_i(t)\right|^2\times\left(X_{\max}-X_{\min}\right)\qquad(19)$$

Na Sl. 1 nacrtan je dijagram toka PPSO.



Start
Generisanje početne populacije:
$k=1:n,\ i=1:N,\ v_{\max,ik}^1=0.5\times(x_{\max,k}-x_{\min,k}),\ t=1,$
$\theta_i^1=U(0.2\pi)$ and $x_{ik}^1=x_{\min,k}+rand_{ik}(x_{\max,k}-x_{\min,k})$
Početna fitnes evaluacija za svaku jedinku jata $f(X_i^1)$ sa
$local\ best\ (pbest)$ i $global\ best\ (gbest)$ pozicijama:
$pbest_i^1=X_i^1$ and $gbest^1=\arg\min_{pbest_i} f(pbest_i^1)$
Ažuriranje brzina i pozicija jedinki k=1:n i i=1:N:
$v_{ik}^t=|\cos_i^t|^{2\sin\theta_i^t}\times(pbest_{ik}^t-x_{ik}^t)+|\sin_i^t|^{2\cos\theta_i^t}\times(\mathrm{g}best_k^t-x_{ik}^t),$
$v_{ik}^t=\min\left(\max\left(v_{ik}^t,\ -v_{\max,ik}^t\right),\ v_{\max,ik}^t\right),$
$x_{ik}^{t+1}=x_{ik}^t+v_{ik}^t,\ x_{ik}^{t+1}=\min\left(\max\left(x_{ik}^{t+1},\ x_{\min,k}\right),\ x_{\max,k}\right).$
Ažuriranje $local\ best\ (pbest)$ i $global\ best\ (gbest)$ pozicija:
$pbest_i^{t+1}=\begin{cases}pbest_i^t, & if\ f(pbest_i^t)\le f(X_i^{t+1})\\ X_i^{t+1}, & \text{otherwise}\end{cases}$
i
$Gbest^{t+1}=\begin{cases}Pbest_i^{t+1}, & if\ f(Pbest_i^{t+1})\le f(Gbest^t)\\ Gbest^t, & \text{otherwise}\end{cases}$
Ažuriranje θ_i i $V_{\max,i}$:
$\theta_i^{t+1}=\theta_i^t+\left(|\cos\theta_i^t+\sin\theta_i^t|\times 2\pi\right),$
$V_{\max,i}^{t+1}=|\cos\theta_i^{t+1}|^2\times(X_{\max}-X_{\min}).$
t=t+1
i=i+1
Ne
t >T?
Da
Optimalno rešenje
Stop

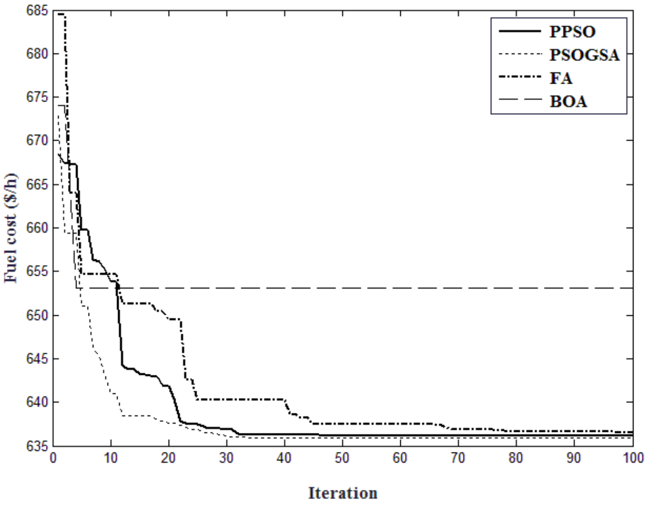
Sl. 1. Dijagram toka PPSO

IV. Rezultati simulacije

Testiranje PPSO algoritma u ovom radu se vrši na standardnom IEEE test sistemu sa 30 čvorova, 6 generatora i ukupnom potrošnjom od 283.4 MW. Uzimaju se u obzir efekat ventila u termoelektranama i gubici snage u sistemu. *B-loss* matrica i koeficijenti troškova i emisije usvojeni su iz [8]. Implementacija PPSO se sprovodi na platformi od 1.6 GHz sa 3 GB RAM primenom MATLAB R2017a. Kao rezultati uzimaju se najbolje vrednosti dobijene posle 30 puštanja algoritma. Veličina dozvoljene greške u (9) je δ = 10^{-6} MW, dok je faktor skaliranja γ_{NOx} jednak 1,000 (\$/t). Minimizacija se vrši sa tri vrednosti težinskog faktora: w = 1 (minimizacija samo troškova goriva), w = 0 (minimizacija samo NO_x emisije) i w = 0.5 (istovremena minimizacija troškova goriva i emisije NO_x gasova). Rezultati dobijeni primenom PPSO upoređuju se sa rezultatima dobijenim pomoću tri sledeća algoritma: (i) hibridnog algoritma koji se sastoji od PSO i gravitacionog pretraživačkog algoritma (eng. PSO - Gravitational Search Algorithm (PSOGSA)) [10], koji je pokazao najbolje rezultate pri rešavanju CEED problema bez uzimanja u obzir efekta ventila [1], [11]; (ii) algoritma optimizacije leptira (eng. Butterfly Optimization Algorithm (BOA)) [12], kao jednog od najnovijih meta-heurističkih algoritama; i (iii) algoritma svica (eng. Firefly Algorithm (FA)) [13], kao jednog od najpoznatijih algoritama.

Konstante testiranih algoritama, koji se primenjuju u simulaciji, date su u Tabeli 1. U Tabeli 2 date su minimalne i maksimalne vrednosti rezultata i njihove standardne devijacije za primenjene algoritme. Iz Tabele 2 sledi da je minimalna vrednost troškova goriva, dobijena primenom PPSO najmanja u odnosu na minimalne vrednosti dobijene pomoću drugih testiranih algoritama. Minimalne vrednosti emisije NO_x gasova su iste u slučaju primene PPSO, PSOGSA i FA. Te vrednosti su bolje (manje) nego u slučaju primene BOA. Standardne devijacije rezultata dobijenih pomoću PPSO su manje nego standardne devijacije rezultata dobijenih pomoću PSOGSA, FA i BOA. U Tabeli 3 date su najbolje vrednosti izlaznih snaga generatora, troškova goriva i emisije gasova, dobijene primenom PPSO za w = 1, w = 0 i w = 0.5.

Na Sl. 3 date su krive konvergencije algoritama PPSO, PSOGSA, FA i BOA algorithms u slučaju minimizacije troškova goriva. Sa Sl. 3 se vidi da PPSO konvergira ka minimalnoj vrednosti za broj iteracija koji je isti kao u slučaju PSOGSA. U poređenju sa FA, PPSO konvergira ka minimalnoj vrednosti za manji broj iteracija. Broj iteracija BOA je manji u odnosu na ostale algoritme ali BOA daje lošije vrednosti minimalnih troškova goriva, emisije gasova i standardne devijacije rezultata. Sl. 3 pokazuje da su početne brzine konvergencije velike za sve primenjene algoritme.



Sl. 3. Krive konvergencije PPSO, PSOGSA, FA i BOA u za slučaj minimizacije troškova goriva.

TABELA 1
KOEFICIJENTI ALGORITAMA KOJI SU TESTIRANI NA STANDARDNOM IEEE TEST SISTEMU SA 30 ČVOROVA I 6 GENERATORA

PPSO		PSOGSA						FA					BOA				
N	T	N	T	G_0	α	C_1	C_2	N	T	A	β_{min}	γ	N	T	c	a	p
50	200	50	200	1	20	0.5	1.5	50	200	0.25	0.2	1	50	200	0.01	0.1	0.8

TABELA 2
MINIMALNE I MAKSIMALNE VREDNOSTI I STANDARDNE DEVIJACIJE, DOBIJENE PRIMENOM PPSO, PSOGSA, FA I BOA NA STANDARDNOM IEEE TEST SISTEMU SA 30 ČVOROVA I 6 GENERATORA

Algoritam		PPSO	PSOGSA	FA	BOA
Minimizacija troškova goriva (w = 1)	Min	635.82129	635.82284	635.83288	640.37240
	Max	647.29186	698.99430	642.65875	663.92341
	SD*	2.376452	18.37740	2.904691	5.989508
Minimizacija emisije NO_x gasova (w = 0)	Min	0.1941785	0.1941785	0.1941785	0.1942077
	Max	0.1941785	0.2195708	0.1941785	0.1966057
	SD*	5.8637e-11	6.23630	1.0606e-10	5.7676e-04
w = 0.5	SD*	2.9438e-2	9.68220	1.96486e-1	2.5305474

* SD označava standardnu devijaciju

V. ZAKLJUČAK

U ovom radu je predložen algoritam PPSO za rešavanje CEED problema. Performanse ovog algoritma pri rešavanju CEED problema su procenjivane na standardnom IEEE test sistemu sa 30 čvorova i 6 generatora. Pri tome, uzimani su u obzir uticaj efekta ventila u termoelektranama i gubici snage u elektroenergetskom sistemu. Zatim su dobijeni rezultati upoređeni sa rezultatima drugih algoritama: PSOGSA koji je u radu [1] pokazao najbolje rezultate pri rešavanju CEED problema na IEEE test sistemu sa 30 čvorova i 6 generatora ali bez uzimanja u obzir efekta ventila; BOA, koji predstavlja jedan od najnovijih meta-heurističkih algoritama; FA, koji je jedan od često primenjivanih algoritama. Poređenjem testiranih algoritama, utvrđeno je da PPSO daje najbolje rezultate: Simulacioni rezultati su pokazali da PPSO ima dobre konvergentne osobine i daje najbolje vrednosti minimalnih troškova goriva u odnosu na algoritme PSOGSA, FA i BOA. Osim toga, utvrđeno je da su standardne devijacije rezultata najmanje u slučaju primene PPSO, da su minimalne vrednosti emisije štetnih gasova iste u slučajevima primene PPSO, PSOGSA i FA i da su one bolje nego u slučaju primene BOA.

TABELA 3
NAJBOLJE VREDNOSTI IZLAZNIH SNAGA, TROŠKOVA GORIVA I EMISIJE GASOVA, DOBIJENE PRIMENOM PPSO

Snaga, MW	$w = 1$	$w = 0$	$w = 0.5$
$P_{s,1}$	5.00000	41.09207	5.00000
$P_{s,2}$	13.44427	46.36641	18.32689
$P_{s,3}$	83.53982	54.44192	79.88927
$P_{s,4}$	74.84721	39.03759	74.81317
$P_{s,5}$	79.79982	54.44609	78.55621
$P_{s,6}$	28.65457	51.54889	28.76874
P_{loss}	1.88568	3.53297	1.95428
Troškovi goriva ($/h)	635.82129	728.66678	638.65784
NO_x (ton/h)	0.226433	0.1941785	0.223048

DODATAK

TABELA 4
B-LOSS MATRICE TEST SISTEMA [8]

Mat-rice	Elementi matrica
B	$\begin{bmatrix} 0.1382 & -0.0299 & 0.0044 & -0.0022 & -0.0010 & -0.0008 \\ -0.0299 & 0.0487 & -0.0025 & 0.0004 & 0.0016 & 0.0041 \\ 0.0044 & -0.0025 & 0.0182 & -0.0070 & -0.0066 & -0.0066 \\ -0.0022 & 0.0004 & -0.0070 & 0.0137 & 0.0050 & 0.0033 \\ -0.0010 & 0.0016 & -0.0066 & 0.0050 & 0.0109 & 0.0005 \\ -0.0008 & 0.0041 & -0.0066 & 0.0033 & 0.0005 & 0.0244 \end{bmatrix}$
B_0	$\begin{bmatrix} -0.0107 & 0.0060 & -0.0017 & 0.0009 & 0.0002 & 0.0030 \end{bmatrix}$
B_{00}	$\begin{bmatrix} 0.00098573 \end{bmatrix}$

TABELA 5
KOEFICIJENTI TROŠKOVA GORIVA I EMISIJE NO_X GASOVA I OGRANIČENJA GENERATORA ZA PRIMENJENI TEST SISTEM [8]

g	a_g	b_g	c_g	d_g	e_g	α_g	β_g	η_g	ξ_g	λ_g	P_g^{min}	P_g^{max}
1	10	200	100	18	3.7	4.091e-2	-5.554e-2	6.490e-2	2.0e-4	2.857	5	150
2	10	150	120	16	3.8	2.543e-2	-6.047e-2	5.638e-2	5.0e-4	3.333	5	150
3	20	180	40	14	4.0	4.258e-2	-5.094e-2	4.586e-2	1.0e-6	8.0	5	150
4	10	100	60	12	4.5	5.326e-2	-3.550e-2	3.380e-2	2.0e-3	2.0	5	150
5	20	180	40	13	4.2	4.258e-2	-5.094e-2	4.586e-2	1.0e-6	8.0	5	150
6	10	150	100	13.5	4.1	6.131e-2	-5.555e-2	5.151e-2	1.0e-5	6.667	5	150

ZAHVALNICA

Ovaj rad je finansijski pomognut od strane Ministarsva prosvete, nauke i tehnološkog razvoja Republike Srbije.

LITERATURA

[1] J. Radosavljević, "A solution to the combined economic and emission dispatch using hybrid PSOGSA algorithm," *Appl. Artif. Intell.* vol. *30*, no. *5*, pp. *445-474*, Jun. 2016.

[2] D. W. Gong, Y. Zhang, C. L. Qi, "Environmental/economic power dispatch using a hybrid multi-objective optimization algorithm," *Int. J. Elec. Power*, vol. *32*, pp. *607-614*, Nov. 2010.

[3] M. Jevtić, N. Jovanović, J. Radosavljivić, D. Klimenta, "Moth swarm algorithm for solving combined economic and emission dispatch problem," *Elektron. Elektrotech.* vol. *23*, pp. *21-28*, Jun. 2017.

[4] M. Gholamghasemi, E. Akbari, A. Rahimnejad, S. E. Razavi, S. Ghavidel, L. Li, "Phasor particle swarm optimization: a simple and efficient variant of PSO," *Soft Comput.* vol. *23*, no. *19*, pp. *9701-9718*, March 2019.

[5] L. Benasla, A. Belmadani, M. Rahli, "Spiral optimization algorithm for solving combined economic and emission dispatch," *Int. J. Elec. Power.* vol. *62*, pp. *163-174*, Nov. 2014.

[6] A. Bhattacharya, P. K. Chattopadhyay, "Solving economic emission load dispatch problems using hybrid differential evolution," *Appl. Soft Comput.* vol. *11*, no. *2*, pp. *2526–2537*, March, 2011.

[7] U. Güvenç, Y. Sönmez, S. Duman, N. Yörükeren, "Combined economic and emission dispatch solution using gravitational search algorithm," *Sci. Iran.* vol. *19*, no. *6*, pp. *1754–1762*, Dec. 2012.

[8] D. Aydin, S. Ozyon, C. Yasar, and T. Liao, "Artificial bee colony algorithm with dynamic population size to combined economic and emission dispatch problem," *Int. J. Elec. Power*, vol. *54*, pp. *144–153*, Jan. 2014.

[9] J. Kennedy, R. Eberhart, "Particle swarm optimization**,"** Proc. ICNN95' International Conference on Neural Networks, IEEE, Perth, WA, Australia, pp. *1942–1948*, 27 Nov. - 1 Dec. 1995.

[10] S. Mirjalli, S. Z. M. Hashim, H. M. Sardroudi, "Training feedforward neural networks using hybrid particle swarm and gravitational search algorithm," *App. Math. Comput*. vol. *218*, no. *22*, pp. *11125-11137*, Jul. 2012.

[11] J. Radosavljević, *Metaheuristic optimization in power engineering*, London, United Kingdom: IET, 2018.

[12] S. Arora, S. Singh, "A hybrid optimization algorithm based on butterfly optimization algorithm and differential evolution," *Int. J. Swarm Intelligence*, vol. *3*, no. **2/3**, pp. *152-169*, Oct. 2017.

[13] X. S. Yang, "Firefly algorithm, L´evy flights and global optimization," in *Research and development in intelligent systems XXVI*, vol. *26*, pp. *209-218*, London, United Kingdom: Springer-Verlag, 2010.

ABSTRACT

Minimization of fuel costs and pollutant emissions in thermal power plants by adjusting electric power outputs from generators represents important problem in power system management. This problem is known as Combined economic emission dispatch (CEED) problem. In this paper, a meta-heuristic algorithm called Phasor particle swarm optimization, which is an improved variant of Particle swarm optimization, is proposed to solve the CEED problem. Parameters of Phasor particle swarm optimization are automatically adjusted during iterations, so this algorithm is adaptive and non-parametric, which is its advantage. The performance of the proposed algorithm for solving CEED problem is evaluated in a standard IEEE test system with 30 nodes and 6 generators. Based on the obtained results, it was determined that this algorithm has better characteristics than the algorithms used in other published papers to solve the CEED problem.

Solving combined economic and emission dispatch problem using Phasor particle swarm optimization

Milena Jevtić, Miroljub Jevtić, Jordan Radosavljević, Sanela Arsić i Dardan Klimenta

Potential of Using Simulated Data in Processing Photoacoustic Measurement Data

M. I. Jordovic Pavlovic, A. D. Kupusinac, S. P. Galovic, D. D. Markushev, M.N. Nesic, K.Lj.Djordjevic, M. N. Popovic

Abstract: This paper explores the potential of using simulated data in calibration of photoacoustic measurement system. The database of simulated experimental values is created using software developed on the bases of the theory-mathematical model. Reliability of the data was gained thanks to the expert knowledge. An artificial neural network as a precise prediction tool is trained on the developed database of simulated data to recognize type of the microphone used as a detector in photoacoustic experiment. The result is classification model satisfies the basic requirements of a photoacoustic experiment: accuracy, reliability and real time operations. The paper discusses the optimization of classification model in terms of used computational power, required precision and process rate in relation with defined problem. The obtained results justify the idea of using simulated data in photoacoustic. Presented theory-mathematical model and classification model are part of developed machine learning framework for processing photoacoustic measurement data.

Keywords: Machine learning, artificial neural networks, simulated data, classification, photoacoustics, microphone

I. Introduction

Machine learning techniques are considered suitable tool for intelligent decision making, and therefore they have found application in various domains. When input and output parameters are linked with some kind of pattern, and sufficient data is available, this pattern can be discovered or approximated by machine learning algorithm being trained on that same data. Subsequently, output for particular inputs outside the learning dataset can be calculated (with more or less accuracy) using this newly discovered pattern. This means that if the quality and the quantity of the data used for learning are sufficient, and the discovered pattern also exists for events that were not part of the learning dataset, the produced result can be used to approximate the outputs based on any future input[1]. Machine learning algorithms, and, in particular, artificial neural networks (ANN), are frequently used as reliable and fast prediction tools. They are often used in photoacoustics (PA), a popular method in photothermal (PT) science in the last few years, for: noise removal in photoacoustic recognition of images [2], simultaneous determination of the laser beam spatial profile and relaxation time of the polyatomic molecules in gases in real time within the trace atmosphere gases monitoring [3][4], reconstruction of optical profile of optically gradient materials based on frequency, magnitude and phase of measured PT response [5], etc.

In this paper, a few of the several results achieved in PA measurement system characterization research are presented. The ultimate goal is material characterization. The aim of the PA measurements is the determination of physical properties (thermal, optical, mechanical, elastic, electronic and other related ones) of the examined structure from its PA response. All PT methods are indirect measurement techniques, and so is the photoacoustics, meaning that these methods are model dependent. In terms of mathematics, obtaining physical properties by these methods is considered an inverse problem that can be assessed in two steps:

1. **Development of the direct (forward) model – direct solution of the inverse problem,** i.e. developing the mathematical model that sufficiently well describes physical processes leading from the optical excitation to the thermal response. First step is theoretical-mathematical modeling of temperature distribution within the sample, on front and back sample surfaces and in its surroundings, and then theoretical-mathematical modeling of the specific PT response (in this case the PA response)
2. **Development of the inverse procedure – inverse solution of the inverse problem,** i.e. the determination of physical properties of the sample based on measured photothermal response, developed mathematical model and well known preset of input parameters (the intensity and modulation frequency of the incident optical radiation). Some of the inverse procedures are fitting, numerical procedures and neural networks. Fitting and numerical procedures are time consuming procedures, demanding the engagement of the researcher. These are drawbacks regarding scientific and further industrial application of the method, where a real time procedure is appreciated. The reasonable choice is

Miroslava Jordović Pavlović is with the Western Serbia Academy of Applied Studies, Užice, Trg Svetog Save 34, 31000 Užice, Serbia (miroslava.jordovic-pavlovic@vpts.edu.rs).
Aleksandar Kupusinac is with Faculty of Technical Sciences, University of Novi Sad, 6 Trg Dositeja Obradovića, 21000 Novi Sad, Serbia (sasak@uns.ac.rs).
Slobodanka Galović is with the Vinča Institute of Nuclear Sciences, 12-14 Mike Petrovića Alasa, 11351 Vinča, Serbia (bobagal@vin.bg.ac.rs).
Dragan Markushev is with the Institute of Physics, University of Belgrade, Pregrevica 118, 11080 Pregrevica, Serbia (dragan.markushev@ipb.ac.rs).
Mioljub Nešić is with the Vinča Institute of Nuclear Sciences, 12-14 Mike Petrovića Alasa, 11351 Vinča, Serbia (mioljub@gmail.com).
Katarina Đorđević is with the Faculty of Physics, University of Belgrade Studentski trg 12, 11000 Beograd, Serbia (katarinaljdjordjevic@gmail.com).
Marica Popović is with the Vinča Institute of Nuclear Sciences, 12-14 Mike Petrovića Alasa, 11351 Vinča, Serbia (maricap@vin.bg.ac.rs)

artificial neural networks as a very efficient machine learning algorithm. Because of the complexity of the inverse problem more than one ANN is needed.

Firstly, the characterization of the sample and the prediction of its thermal, mechanical and optical properties based on its PA response, require the use of one ANN. This is already done in [6]. But the necessary precondition is that the PA response in use is influenced only by the sample, which, unfortunately, is not the case. The PA response is non- linearly affected not only by the sample but also by the measurement instrument chain and the appearing noise. So, the PA response has to be corrected first, in order to obtain the so called "true" signal. In the first preprocessing procedure that was developed, noises are removed during data acquisition. In the second preprocessing procedure, calibration of the measurement system has to be done. Because of the dominant impact of the microphone on the distortions in the measurement instrument chain, as the consequence of using minimum volume cell configuration of the PA experimental set-up, calibration of the measurement system boils down to the calibration of the microphone. The key of this brand new idea is the determination of microphone transfer function. Furthermore, the division of PA response amplitude data by corresponding microphone characteristics and its subtraction from the PA response phase data will result in gaining the so called "true" signal, originating only from the sample. Unfortunately, microphone specifications provided by the manufacturer are not precise enough, particularly in the case of phase transfer function. Besides, microphone cavity is not considered as the source of resonances, which is inevitable in PA measurements. Since, these specifications could not be used, the other solution is needed. Non-linear influence of the microphone on a PA response suggests ANN application. Having in mind that ANN seeks large datasets (is data hungry) [7], the first requirement for the application of neural networks is set. But this requirement is opposed to two facts related to PA measurements: firstly, such a numerous experimental collection is very difficult to obtain, and secondly, based on the experience, real experimental data can hide a very serious problem of the influence of the measurement system on the estimated parameter values [8]. Therefore, another solution for database creation is presented: the idea of theoretical-mathematical model as a base for designing a software for the simulation of PA experimental values. Thanks to the developed software, amplitude and phase data of the simulated PA response are obtained. Here, satisfactory credibility to the experiment is of essential importance in order to make the newly created method precise enough. Therefore, expert knowledge (i.e., the preset input parameters) is crucial for the solution of this problem.

Simulated data have often been used for training in machine learning problems in the past few years [9][10][11], but as far as we know, the idea of using simulated experimental values, obtained by developed software based on a theoretical - mathematical model, for training a machine learning model is new. This article presents a few steps of a complex correction procedure performed in photoacoustic measurements. Firstly, a complete method of making a large amount of reliable simulated data as a precondition for applying neural networks as the inverse solution of the inverse problem is explained. Secondly, a process of designing classification model for microphone type recognition as the first step in recognizing measurement system characteristics is discussed on the base of optimal computational complexity, required precision and process rate in relation to the given problem and available data set. Classification of microphone type will determine the shape of the transfer function and the levels of signal exaggeration and attenuation. Once the class of the microphone is defined, characterization of microphone will be simplified by limiting the database made for various types of microphones to a database of a particular microphone type. This idea is presented in our previous work [12]. That way, time, and computational power are saved, which are real benefits of the classification model. Learning on the defined database of classified microphone type, ANN based model for microphone characterization [13] predicts characteristic microphone parameters with satisfying accuracy, which together with the corresponding shape, precisely determine microphone transfer function [12].

This paper shows that if a massive dataset is obtained and the quality of data is high, less computation power is needed, and higher process rate is gained for the solution of machine learning problem.

II. Theoretical -Mathematical Model of Photoacoustic Response

Photoacoustics, as one of photothermal methods, is based on the photothermal effect. The photothermal effect is the effect of generation of heat as a consequence of the absorption of the incident electromagnetic radiation, from a wide spectrum of wavelengths, in different relaxation and de-excitation processes. This way generated heat causes the disruption of the thermodynamic state of the sample (pressure, temperature, density) which propagates through the sample and the nearby environment, producing a number of detectable phenomena. In photoacoustics, the first and the most used photothermal method, a sample is placed inside the photoacoustic cell that contains air and microphone. It is exposed to a modulated light beam which causes periodic sample heating. As a consequence, the air pressure in the PA cell oscillates, which can be detected by a microphone [14].The photoacoustic cell can be designed in a so called "reflection configuration", with the source and the microphone set up on the same side of a sample, or the "transmission configuration", where a sample is placed between the source and the microphone. In our experimental set up, the minimum volume cell configuration is employed. It is kind of transmission cell configuration where sample is mounted directly on the top of the microphone, instead of the dust cover, as presented in figure 1, [15]. This way, the microphone chamber acts as the PA cell, closed by the sample on one side and the microphone diaphragm on the other one, which causes disruptions of the recorded signal on its endings [16].

Power levels of experimentally recorded signals are, generally, low. In order to make the level of the recorded signal higher than the level of the noise (real – flicker noise, coherent signal deviation and random noise), the absorption of the sample has to be large. In the case of materials with significant reflection, the additional coating is needed, while in the case of transparent (or semitransparent) samples, the coefficient of transmission has to be augmented. However, due to high level of transmission, the absorption of the incident radiation in the surrounded air can`t be neglected and the recorded signal begins to contain unnecessary information. The problem is even bigger in the case of the minimum volume cell configuration, where the microphone has to be protected because of the small dimensions of the cell. Another solution is, also, an additional layer of high absorption.

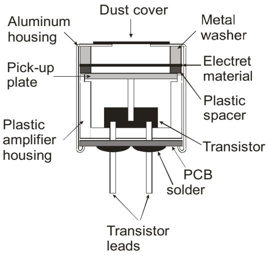


Fig. 1. Experimental setup

Photoacoustic response within the transmission configuration is the sum of two dominant signal components: thermoconducting and thermoelastic component. Thermoconducting component arises due to the periodic heat flow from the sample to the surrounding gas (thermal-piston effect) and thermoelastic component arises due to the thermoelastic banding of the sample (drum effect) [17][18][19][20][21][22][23][24][25].

In our experiments two-layer structure is employed. The first layer is black coating, and the second layer is the investigated sample. Theoretical-mathematical model of PA response of a two-layer system, used for obtaining the dataset, is given by following expressions [26][27]:

$$\tilde{p}_{tot} = \tilde{p}_{th} + \tilde{p}_{ac} \tag{1}$$

$$\tilde{p}_{th} = \delta P = \frac{\gamma P_0}{l_g T_0}\frac{1}{\widetilde{\sigma_g}}\tilde{\vartheta}(l_s) \tag{2}$$

$$\tilde{p}_{ac} = \frac{3\gamma P_0}{l_a}\frac{R^2}{l_s^3}\left[\alpha_{T1}\int_0^{l_1}\left(x-\frac{l_s}{2}\right)\tilde{\vartheta}(x)dx + \alpha_{T2}\int_{l_1}^{l_s}\left((x-l_1)-\frac{l_s}{2}\right)\tilde{\vartheta}(x)dx\right] \tag{3}$$

Where p_{tot} is total pressure that we want to record by photoacoustic, p_{th} is the thermoconducting component and p_{ac} is the thermoelastic component. Furthermore P_0 is the presser in the cell, V_0 is the volume of the cell (in the case of the minimum volume cell V_0 represents the volume of the chamber cavity), γ represents the heat capacity ratio, α_T is the thermal expansion coefficient, R_c is the radius of the chamber in front of the microphone diaphragm, $l_1$and l_2 are the thicknesses of the first and second layer, while l_s is the sum of the thicknesses these two layers ($l_1$and l_2). $\vartheta(x)$ represents temperature variations inside the samples and $\vartheta(l_s)$ is the surface temperature variation on the rare surface. Expressions for these temperature variations are given in the article [28][29]. The presented model described the total presser as photoacoustic response, and its components in the two-layer system surrounded by the air and it is based on the Generalized model of heat conduction that implies finite heat propagation speed. The system depicts volume absorption of incident optical beam in both layers [26][27][28][29].

Appearance of amplitude and phase characteristics of the theory-mathematical simulated total pressure are shown on figure 2a) and 2b) respectively.

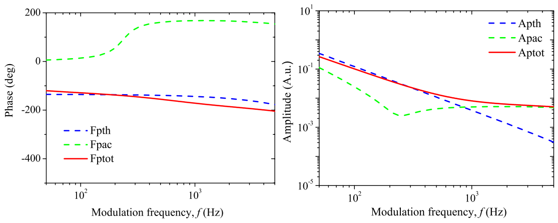


Fig. 2 Simulated amplitude and phase (solid line) of the total photoacoustic signal, $p_{tot}(f)$, as a function of the modulation frequency f, together with the appropriate components $p_{th}(f)$ and $p_{ac}(f)$ (dotted lines).

In a minimum volume cell PA experiment [30], microphone is the fundamental part of the detector system. Microphone is an acoustic-electric converter, but its transfer function in frequency and time domain differ due its construction, applied geometry and membrane type. In the literature [8] and in our experimental experience, microphone behavior is described as filtering. At low frequencies (< 1 kHz), electret microphones (commonly used in PA) usually act as electronic high-pass filters, while at high frequencies (> 1 kHz) these microphones usually act as acoustic low-pass filters.

The influence of the measurement chain, including the microphone as the component that has the greatest impact in signal distortion, is given by the following mathematical expressions describing total transfer function:

$$H_1^e(f) = \frac{1}{1 - j\frac{f_1}{f}} \tag{4}$$

$$H_2^e(f) = \frac{1}{1 - j\frac{f_2}{f}} \tag{5}$$

$$H^a(f) = \frac{f_3^2}{f_3^2 - f^2 + jff_3\xi_3} + \frac{f_4^2}{f_4^2 - f^2 + jff_4\xi_4} \tag{6}$$

$$H_{mic}(f) = H_2^e(f)H_{total}^a(f) \tag{7}$$

$$H_{total}(f) = H_1^e(f)H_{mic}(f) \tag{8}$$

In previous equations, $H_1^e(f)$ represents electronic characteristic of the influence of the other components in the measurement chain, first of all the sound card, and f_1 is the characteristic frequency that describes this system. Based on experimental experience, it is assumed that this frequency is constant. $H_2^e(f)$and $H^a(f\ )$ represent electronic and acoustic characteristics of the microphone. f_2 correspondes to the characteristic frequency of the electronic high-pass filter and f_3 and f_4 to the characteristic frequencies of the acoustic low-pass filters of the microphone, ξ_3 and ξ_4 are reciprocial values of the quality factor, or, in other words, the double value of the damping factor. The product of these two components represents the microphone response. As a consequence, the microphone response in frequency domain is deviated in amplitude and phase, especially at the begging and at the end of frequency range. Different microphone types have different transfer functions, but transfer functions of two microphones of the same type are usually different, because, in practice, two identical microphones do not exist. Theoretical-mathematical model for the total photoacoustic signal recorded by the minimum volume cell photoacoustic experimental set up represents product of the total pressure and the total transfer function:

$$S(f) = \sigma p_{total}(f) H_{total}(f) \tag{9}$$

Based on this equation and numerical simulations of the experiments, the database is obtained. Amplitude and phase data of the simulated PA response are given in figure 3a) and 3b). All the curves (amplitude and phase) of distorted photoacoustic signal have expected shape, according to experimental experience. There are no outliers.

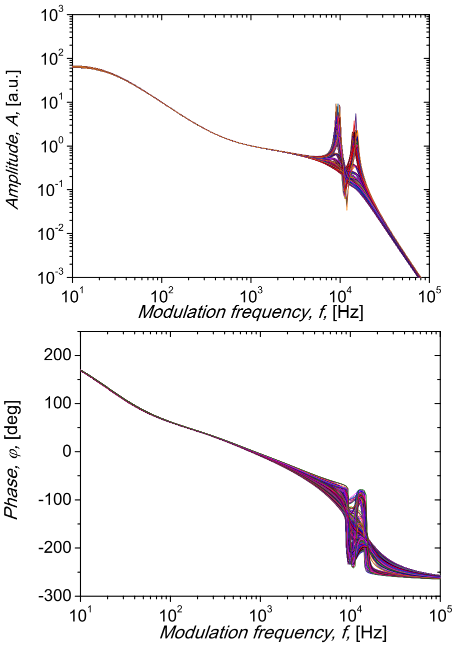


Fig. 3 Curves a) amplitude and b) phase of distorted photoacoustic signals with different microphone characteristics from the dataset used for network training[12]

III. Database description

Based on theoretical-mathematical model, software for creating simulated experimental values or numerical experiments is designed using programming IDE of Matlab. Microphone theoretical characteristics, corresponding to commercially available microphones ECM30B, ECM60 and WM66, are given in Table 1. Beside these microphone types, frequently used in PA experiments, simulations for another type of microphone are created, the so called ideal microphone (IM). Considering ideal microphone is of great importance for the correction procedure. If a microphone exerted ideal behavior, meaning it had flat PA response, that would mean that measurement chain would be equally sensitive in the whole frequency domain, so the correction procedure would be unnecessary. So, taking IM into account, we are saving the time.

TABLE I

THEORETICAL VALUES FOR ALUMINUM SAMPLE

	Dye	Aluminum
Thermal conductivity [$Wm^{-1}K^{-1}$]	70	210
Thermal diffusivity [m^2s^{-1}]	$2.5*10^{-5}$	$8.6*10^{-5}$
Thermal relaxation time [s]	10^{-4}	10^{-12}
Absorption coefficient [m^{-1}]	10^8	$145*10^6$

During the process of the examination, the black dye-aluminum structure was investigated. Aluminum plate, 197 µm in thicknesses and with radius of 10 µm was covered in black ink dye, 2 µm in thicknesses. Thermal, thermal memory and optical parameters used for obtaining database are given in Table 1.

Expert knowledge was crucial in obtaining similarity good enough with the experiment. Based on experimental experience, characteristic microphone parameters are considered to have different stability, regarding the reproducibility in each measurement. Accordingly, different value ranges were set for different parameters. Frequency f_2 is the most stable parameter due to its origin from RC microphone characteristic, so three values ware taken for network training: central value $f_{20} = 25$ Hz for the microphone ECM30B and two values which are ± 5 % apart from the central value (23.75 Hz and 26.25 Hz). By analogy, the values for the ECM60 are: 14.25Hz, 15Hz and 15.75 Hz, for the WM66 they are: 61.75 Hz, 65 Hz and 68.25 Hz, while for IM the values are: 0.475Hz, 0.5 Hz and 0.525 Hz. Frequencies f_3 and f_4 are more dependent on experimental conditions then f_2, so they are less stable than f_2. Ten values, equally distanced in the corresponding ranges, were considered to be good enough for the description of experimental behavior related to those two frequencies. f_{30} is taken in the range 8930-9866 Hz and f_{40} is taken in the range 13965-15432 Hz for ECM30B. Microphones ECM60 and WM66 have the same ranges for frequencies f_3 and f_4, 7980-8817 Hz and 13015-14383 Hz respectively. For IM f_3 is in the range of 190000-209998 Hz while f_4 is in the range of 285000-314997 Hz respectively. Damping factors of the second order low-pass filter ξ_3 and ξ_4 are strongly dependent on experimental conditions and they are the most unstable parameters. Each value range, for ξ_3 and ξ_4, was chosen based on the peak appearing in the amplitude characteristic of the second order filter. Critical value of quality factor in the case of limitary situation where signal is extremely damped and respectively unforced is Q=0.5. Significant change happens from Q=1 to Q=100 hence $\xi \in [0.99,0.015]$. 15 values, irregularly distributed in this range, were taken for the each type of microphone. This kind of microphone parameter distribution was assumed to be good enough to simulate all possible experimental situations. The discussion and comparison of inverse problem-solving concepts in photoacoustics is presented in our previous work [31]. There are 65,000 paired curves for each microphone type, as 65,000 simulated experimental results, and those are 65,000 records of the database. Paired curves (two curves) mean that there are both amplitude and phase data for the given set of microphone parameters. Each curve contains data sampled at 200 frequency values in the range from 10Hz to 100kHz. By taking such a wide frequency domain, the possibility of using microphones with different membrane material (mylar, nickel, graphene) is considered. In total, every record is represented with 400 samples, 200 samples of amplitude and 200 samples of phase characteristics. Those are features for our machine learning problem. In other words each frequency is presented with two features, sample of amplitude and sample of phase, so we have resolution of two for every point on frequency axes. At the end of each database record, the information about which microphone type a particular record belongs to is written. The classification problem has 4 classes of microphones, symbolically presented with 0, 1, 2 and 3.

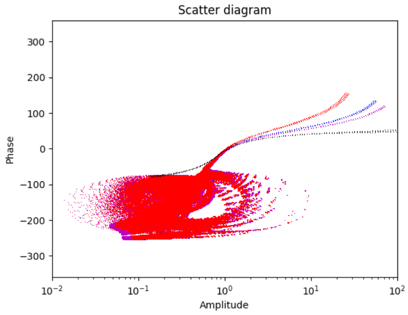


Fig.4. Visualization of the data, different colors correspond to different microphone types

Visualization of the data used in classification modeling, the form of scatter diagram, is given in Fig. 2. Each point on a scatter diagram is one point of 200 points that corresponds to one curve of 270,000 curves in the database. Different classes of microphone are presented with different colors. Analyzing the diagram, one can conclude that points are completely classified to four classes or four microphone types in upper-right part of the diagram, meaning for certain distribution of amplitude and phase values it is clear to which class point belongs. That distribution of amplitude and phase values are happening in a low frequency domain. In lower-left part of the diagram points are mixed, meaning that for that distribution of amplitude and phase values it is not clear to which class point belongs, i.e. curves (or classes) overlap. Thus, classification model has more difficult task because of the overlap. Training, validation and test sets are obtained randomly because dataset is first shuffled and then divided into training, validation and test set. Generalization of the results is obtained on that way, thus 243 000 records or 90% of the total number of records belongs to the training set, 13500 records or 5% belong to the validation set and the rest belongs to the test set.

IV. Research results and discussion

Once, the topology of the model is chosen, the next step is fine-tuning of topology itself, parameters and hyperparameters of the model. It is done in iterative process idea-code-experiment, with a numerous attempt using literature suggestions [32][33] and experience.

In pre-processing step, data scaling was done by performing the normalization of the input and output. Max normalization was chosen. It means that each element x_i of the input vector is divided by its maximum absolute value, which is the maximum of absolute values of all the samples, a total of 270000 values, at the i-th frequency. In other words, it is absolute maximum value of the i-th row of the input matrix. This way normalization of the input vector is done, all the values of the input vector are equal or less than unity. Similarly, normalization of the output vector is done. For weights parameters initialization, among others Xaviar algorithm [34] is chosen. The activation function *tanh()* is used for forward propagation and the Adam algorithm [35] is used for the optimization of weights in backpropagation. The optimization is intensified by the Mini-batch technique, size of 128. Because of the classification function softmax in the last layer, a cross entropy with logits is used as the error function and system performance measure during training. Neural network tuning on number of hidden layers and the number of neurons is presented in Table 2.

TABLE II

NUMBER OF HIDDEN LAYERS AND NUMBER OF NEURONS IN HIDDEN LAYER(S) ANALYSES

No.of hidden layers	1	2	2	2	1	2	2	1	2	2	1	2	1
No. of neurons of the 1. h. l.	10	8	7	9	5	3	4	4	2	3	3	2	2
No. of neurons of the 2. h. l.	/	2	3	1	/	2	1	/	2	1	0	1	0
Train accuracy(%)	99.99	99.99	99.99	75.02	99.99	99.99	50.03	99.99	99.99	81.15	99.99	75.01	99.99
Dev accuracy(%)	99.99	99.99	99.99	74.15	99.99	99.99	49.19	99.99	99.99	82.09	99.99	75.03	99.99
Test accuracy(%)	99.99	99.99	99.99	75.45	99.99	99.99	50.15	99.99	99.99	81.11	99.99	74.7	99.99
Number of epochs	100	100	100	100	100	100	100	100	100	100	100	100	100
Prediction time (ms)	14.34	17.89	17.44	/	14.06	16.75	/	13.89	16.89	/	13.73	/	13.93

According to Table 2, for the defined classification problem and the dataset of 270000 records following conclusions can be drawn. One neuron in second hidden layer in configuration of two hidden layers is not appropriate and those topologies were dismissed, but 2 neurons in second hidden layer are satisfying. The reason are 4 classes at the output. There is no difference in accuracy in the case of the configuration with one hidden layer and in the case of configuration with two hidden layers with same total number of neurons. Based on experimental experience one can say that for other machine learning problems that was not a case. This is specificity of this particular problem. So, the topology and the choice of model parameters and hyperparameters are singularity of machine learning problem and the quantity and quality of available data. Minimum configuration that satisfies required accuracy is one hidden layer with 2 neurons. It is surprisingly small number of neurons, which can be justified with the large data set. It means that learning with large datasets decreases the number of computational units of ANN configuration, it becomes computationally simpler. Large dataset brings into the model huge knowledge about the problem, in the case of our classification problem knowledge about photoacoustic experiment environment. Using this knowledge ANN needs less computational power and less epochs for learning. Analyzing the obtained prediction time of different topologies of classification model, the most important influence on the processing rate has the number of hidden layers, the number of neurons in layer has minor influence, even if there is significant difference in the number of neurons.

Concerning the prediction, the network gives very high accuracy, train, dev and test accuracy are equal, 99.99%. Concerning the training, the network obtained good results even for very quick training, that lasts 100 epochs. According to the equal values of training, dev and test accuracy and low error function on the new data sets we can conclude that the network generalizes very well. There is no overfitting.

The reliability of the model was tested on simulated data. Sixteen different independent datasets, meaning four different amplitude and phase characteristics for each type of microphone were created, where the microphone parameter values differed from those on which the network was trained, but in the given parameter range. Results are presented in Table 3. According to Table 3 our model is reliable, it recognizes the microphone type precisely and gives an answer regarding the microphone type in real time.

Results of the model on real experimental data are presented in [12].

TABLE III
RESULTS OF INDEPENDENT TESTS

Test	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Class.	1	3	0	2	1	2	3	0	2	3	1	2	1	0	3	0
Accuracy	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

V. Conclusion

In this paper a complete explanation of the necessity for simulation data in the processing real photoacoustic measurement data is given. Software for simulations is designed based on the presented theoretical-mathematical model, while the credibility to the experiment is obtained using expert knowledge. Classification model for microphone type recognition is trained on the obtained database. Because of the huge, reliable dataset, knowledge about the photoacoustic experiment is embedded in the classification model so it could be optimized to a pretty simple topology, while the learning process was extremely efficient. In terms of precision and real time processing, classification model satisfies requirement of the photoacoustic experiment. In terms of reliability, classification model did not make any mistake in tests maintained with simulated data. The benefits of the presented model for PA measurements are multiple. By recognizing the microphone type the shape of transfer function and levels of signal exaggeration or attenuation are determined and that will simplify the further procedure of recognition of microphone characteristics in order to deprive PA signal of instrumental deviations. If the recognized microphone has flat characteristics the correction procedure is skipped. In the case of shaped response the correction procedure is done using only database of recognized microphone instead of whole database for all types of microphone. The processing time is saved this way. The generality of the model could be accomplished by extending the number of microphone types if such requirement of the experiment exists. In the future, we intend to explore modeling of noise distribution to generate data similar to real data and skip the first step of correction procedure, the noise removal.

References

[1] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data*. AML Book, 2012.

[2] D. Allman, A. Reiter, and M. A. L. Bell, “Photoacoustic Source Detection and Reflection Artifact Removal Enabled by Deep Learning,” *IEEE Trans. Med. Imaging*, vol. 37, no. 6, pp. 1464–1477, 2018.

[3] M. Lukić, Ćojbašić, M. D. Rabasović, D. D. Markushev, and D. M. Todorović, “Laser Fluence Recognition Using Computationally Intelligent Pulsed Photoacoustics Within the Trace Gases Analysis,” *Int. J. Thermophys.*, vol. 38, no. 11, 2017.

[4] M. Lukić, Ž. Ćojbašić, M. D. Rabasović, and D. D. Markushev, “Computationally intelligent pulsed photoacoustics,” *Meas. Sci. Technol.*, vol. 25, no. 12, p. 125203, 2014.

[5] M. N. Popovic, D. Furundzic, and S. P. Galovic, “Photothermal Depth Profiling Of Optical Gradient Materials By Neural Network,” *Publ. Astron. Obs. Belgrade*, vol. 89, no. May 2015, 2010.

[6] S. P. Djordjevic, K.Lj., Markushev, D.D., Ćojbašić, Ž. M., Galović, “Photoacoustic measurements of the thermal and elastic properties of n-type silicon using neural networks,” *Silicon , Springer*.

[7] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] S. Aleksic, D. Markushev, D. Pantic, M. Rabasovic, D. Markushev, and D. Todorovic, “Electro-acustic influence of the measuring system on the photoacoustic signal amplitude and phase in frequency domain,” *Facta Univ. - Ser. Physics, Chem. Technol.*, vol. 14, no. 1, pp. 9–20, 2016.

[9] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla., “SceneNet: Understand- ing real world indoor scenes with synthetic data.,” in *CVPR*, 2015.

[10] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from Simulated and Unsupervised Images through Adversarial Training,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2242–2251.

[11] G. Cosne *et al.*, “Using Simulated Data to Generate Images of Climate Change,” in *ML-IRL workshop at ICLR*, 2020, pp. 1–9.

[12] M. I. Jordovic-Pavlovic *et al.*, “Computationally intelligent description of a photoacoustic detector,” *Opt. Quantum Electron.*, vol. 52, no. 5, pp. 1–14, 2020.

[13] M. I. Jordović-Pavlović, M. M. Stanković, M. N. Popović, Ž. M. Ćojbašić, S. P. Galović, and D. D. Markushev, “The application of artificial neural networks in solid-state photoacoustics for the recognition of microphone response effects in the frequency domain,” *J. Comput. Electron.*, vol. 19, no. 3, pp. 1268–1280, 2020.

[14] A. Rosencwaig, “Photoacoustic Spectroscopy of Solids,” *Opt Commun, 7(4)*, pp. 305–308, 1973.

[15] L. C. M. Perondi, L. F, Miranda, “Minimal-Volume Photoacoustic Cell Measurement of Thermal Diffusivity: Effect of the Thermoelastic Sample Bending.,” *J. Appl. Phys.*, vol. 62, pp. 2955–2959, 1987.

[16] M. Popovic, M. Nesic, S. Ciric-Kostic, M. Zivanov, D.Markushev, M. Rabasovic, S. Galovic, “Helmholtz Resonances in Photoacoustic Experiment with Laser-Sintered Polyamide Including Thermal Memory of Samples,” *Int. J. Thermophys.*, vol. 37, no. 12, pp. 1–9, 2016.

[17] A. Rosencwaig and A. Gerscho, “Photoacoustic Effect with Solids: A Theoretical Treatment,” *Science (80-.).*, vol. 190, pp. 556–557, Nov. 1975.

[18] F. A. McDonald and G. C. Wetsel, “Generalized theory of the photoacoustic effect,” *J. Appl. Phys.*, vol. 49, no. 4, pp. 2313–2322, Apr. 1978.

[19] L. Rousset, F. Lepoutre, and L. Bertrand, “Influence of thermoelastic bending on photoacoustic experiments related to measurements of thermal diffusivity of metals,” *J. Appl. Phys.*, vol. 54, no. 5, pp. 2383–2391, 1983.

[20] P. M. Nikolic and D. M. Todorović, “An investigation of semiconducting materials using a photoacoustic method. u: Technical sciences book 40, Monographs, Belgrade: Serbian Academy of Sciences and Arts Department, vol. DCXLVIII.” Serbian Academy of Sciences and Arts, 2001.

[21] D. D. Markushev, M. D. Rabasović, M. V Nesic, M. N. Popovic, and S. P. Galovic, “Influence of thermal memory on thermal piston model of photoacoustic response,” *Int. J. Thermophys.*, vol. 33, no.

10–11, pp. 2210–2216, 2012.

[22] M. V Nesic, S. P. Galovic, Z. N. Soskic, M. N. Popovic, and D. M. Todorović, “Photothermal thermoelastic bending for media with thermal memory,” *Int. J. Thermophys.*, vol. 33, no. 10–11, pp. 2203–2209, 2012.

[23] S. P. Galovic, Z. N. Soskic, M. N. Popovic, Z. Stojanovic, D. Cevizovic, and Z. Stojanovic, “Theory of photoacoustic effect in media with thermal memory,” *J. Appl. Phys.*, vol. 116, no. 2, pp. 0–12, 2014.

[24] D. M. Todorović, M. D. Rabasovic, and D. D. Markushev, “Photoacoustic elastic bending in thin film—Substrate system,” *J. Appl. Phys.*, vol. 114, no. 21, p. 213510, 2013.

[25] D. M. Todorović, M. D. Rabasovic, D. D. Markushev, and M. Sarajlić, “Photoacoustic elastic bending in thin film-substrate system: Experimental determination of the thin film parameters,” *J. Appl. Phys.*, vol. 116, no. 5, 2014.

[26] M. Popovic, *Generalizovani fotoakustični odziv dvoslojnih struktura*. Beograd: Zadužbina Andrejević, 2018.

[27] M. N. Popovic, “Fotoakustički odziv transmisione fotoakustičke konfiguracije i analiza rezonantnih fenomena za dvoslojne uzorke sa toplotnom memorijom,” Univerzitet u Novom Sadu, 2016.

[28] M. N. Popovic, D. D. Markushev, M. V. Nesic, M. I. Jordovic-Pavlovic, and S. P. Galovic, “Optically induced temperature variations in a two-layer volume absorber including thermal memory effects,” *J. Appl. Phys.*, vol. 129, no. 1, 2021.

[29] M. Popovic, M. Nesic, M. Zivanov, D. Markushev, and S. Galovic, “Photoacoustic response of a transmission photoacoustic configuration for two-layer samples with thermal memory,” *Opt. Quantum Electron.*, vol. 50, p. 330, 2018.

[30] M. D. Rabasovic, M. G. Nikolic, M. D. Dramicanin, M. Franko, and D. D. Markushev, “Low-cost, portable photoacoustic setup for solid samples,” *Meas. Sci. Technol.*, vol. 20, no. 9, p. 95902, 2009.

[31] M. Nesic *et al.*, “Development and comparison of the techniques for solving the inverse problem in photoacoustic characterization of semiconductors,” *Opt. Quantum Electron.*, vol. 53, no. 7, p. 381, 2021.

[32] Q. V Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, “On Optimization Methods for Deep Learning,” in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 265–272.

[33] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 LECTU, pp. 437–478, 2012.

[34] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, vol. 9, pp. 249–256.

[35] D. P. Kingma and J. Ba, “Adam: {A} Method for Stochastic Optimization,” *CoRR*, vol. abs/1412.6, 2014.

Genetic Algorithm for Bent Functions Generating

Milan Stojanović and Suzana Stojković

Abstract— **The importance of unique bent functions (most significantly in cryptography) creates a demand for their generation. Bent function generation is an interesting problem and, in this paper, we explore the idea of using invariant spectral operations in a Genetic algorithm for generating bent functions. Invariant spectral operations, when executed on bent function, resulting function is also bent. If multiple operations are performed consecutively, then there is a possibility that the newly generated bent function is not unique. A genetic algorithm is used to search the solution space in order to produce the most unique bent functions, for the least number of invariant spectral operations.**

Index Terms— **Bent functions, invariant spectral operations, genetic algorithm.**

I. Introduction

Bent functions are Boolean functions most distant from affine functions. They were introduced by O.S. Rothaus in 1976. [1], and they have characteristics that are interesting for cryptographic applications. There are many algorithms for the generation of the bent function, see for example [2-9] and references therein.

A very important characteristic of the bent functions is flat Walsh spectra. All Walsh spectral coefficients of n-variable bent functions have the same absolute value equal to $2^{n/2}$. Invariant spectral operations are operations that do not change the absolute values of spectral coefficients, i.e., they only permute or change the sign of spectral coefficients. It follows that new bent functions can be generated from any known bent function by applying invariant spectral operations. References [7-9] elaborate methods for bent functions generation by using invariant spectral operations. The main disadvantage of those methods is that the same bent function can be generated by applying different sequences of operations.

Genetic algorithm is inspired by natural selection, that belongs to the evolutionary algorithm group. This algorithm is used to optimize a solution for a corresponding problem. It can be used most effectively when the search space is vast, but the solution does not need to be perfect, only optimal to some degree.

This paper proposes the usage of a Genetic algorithm for the generation of bent functions. Bent functions belong to the vast space of Boolean functions. Therefore, the search for unique bent functions can be presented as executing a sequence of invariant spectral operations, and optimization is used in the sequence of operations, so that we will produce as many different bent functions as possible.

The paper is organized in the following way: Section II presents the ANF representation of bent function. Section III covers Invariant spectral operations. Oscar-Bent functions are presented in Section IV and the Genetic algorithm is defined in Section V. Section VI explains the problem definition and usage of the Genetic algorithm for the generation of bent functions. Section VII goes over the results, and Section VIII gives a conclusion.

II. ANF Representation of Bent Functions

A. Definition

An n-variable Boolean function $f(x_1, x_2, \dots, x_n)$ can be presented by the algebraic normal form (ANF), or the positive polarity Reed-Muller expansion as:

$$f(x_1, \dots, x_n) = \sum_{i=0}^{2^n-1} S(i) \prod_{k=0}^{n-1} x_k^{i_k}$$

where $S(i)$ is the Reed-Muller spectral coefficient and $i_0 i_1 \dots i_{n-1}$ is the binary representation of the index i.

Reed-Muller spectral coefficients of bent functions are equal to 0 for each input vector with the number of ones greater than $n/2$. The maximal number of variables in a product term is called the degree of f [8].

B. Disjoint quadratic function

The disjoint quadratic function contains $n/2$ disjoint quadratic terms, defined as:

$$f(x_1, \dots, x_n) = x_1x_2 \oplus x_3x_4 \oplus \dots \oplus x_{n-1}x_n$$

III. Invariant spectral Operations

A. Definition

Invariant spectral operations do not change the absolute values of Walsh spectral coefficients, they only permute or change the sign of spectral coefficients. These changes preserve the flat spectrum.

Due to the simplicity of invariant spectral operations in the Reed-Muller domain, all operations are introduced in this domain. For consistency, all examples will be provided starting

Milan Stojanović is master's-degree student in Computer Science at Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mail: milances.14@gmail.com)

Suzana Stojković is with Dept. of Computer Science, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14 18000 Niš, Serbia (e-mail: suzana.stojkovic@elfak.ni.ac.rs).

from the Disjoint quadratic function for $n = 6$.

B. Function complement

Function complement is defined as:

$$f_2 = \overline{f_1} = f_1 \oplus 1$$

For example, if

$$f_1(x_1, x_2, x_3, x_4, x_5, x_6) = x_1x_2 \oplus x_3x_4 \oplus x_5x_6$$

The resulting function will be:

$$f_2(x_1, x_2, x_3, x_4, x_5, x_6) = x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \oplus 1$$

C. Variable complement

Variable complement replaces the input variable i by its complement $x_i' = x_i \oplus 1$.

If variable complement on variable x_4 is performed, the function f_1 is transformed to:

$$\begin{aligned} f_2(x_1, x_2, x_3, x_4, x_5, x_6) &= f_1(x_1, x_2, x_3, \overline{x_4}, x_5, x_6) \\ &= x_1x_2 \oplus x_3(x_4 \oplus 1) \oplus x_5x_6 \\ &= x_1x_2 \oplus x_3x_4 \oplus x_3 \oplus x_5x_6 \end{aligned}$$

D. Disjoint spectral translation

Disjoint spectral translation replaces the input variable i by $x_i' = x_i \oplus x_j$, where $i \neq j$.

In the given example, if x_3 is replaced by $x_3 \oplus x_6$, following function is generated:

$$\begin{aligned} f_2(x_1, x_2, x_3, x_4, x_5, x_6) &= f_1(x_1, x_2, x_3 \oplus x_6, x_4, x_5, x_6) \\ &= x_1x_2 \oplus (x_3 \oplus x_6)x_4 \oplus x_5x_6 \\ &= x_1x_2 \oplus x_3x_4 \oplus x_4x_6 \oplus x_5x_6 \end{aligned}$$

E. Spectral translation

In the general case, we can define spectral translation as adding linear member x_i to the function:

$$f_2 = f_1 \oplus x_i$$

If in our example x_2 is added, resulting function is:

$$f_2(x_1, x_2, x_3, x_4, x_5, x_6) = x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \oplus x_2$$

F. Permutation of variables

Permutation of variables is defined as the interchange of two input variables $x_i \leftrightarrow x_j$, where $i \neq j$.

$$f_2(x_1, \dots, x_i, \dots, x_j, \dots, x_n) = f_1(x_1, \dots, x_j, \dots, x_i, \dots, x_n)$$

In the given example if we interchange input variables x_3 and x_6 the resulting function is:

$$f_2(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_6, x_4, x_5, x_3)$$

$$f_2(x_1, x_2, x_3, x_4, x_5, x_6) = x_1x_2 \oplus x_6x_4 \oplus x_5x_3$$

G. Generalized spectral translation

The generalized spectral translation is defined for the function f which has n variables ($n = 2 * k, k \geq 3$) and contains $n/2$ disjoint quadratic terms:

$$f(x_1, .., x_n) = \cdots x_{i_1}x_{j_1} \oplus x_{i_2}x_{j_2} \oplus \dots \oplus x_{i_{n/2}}x_{j_{n/2}}$$

Performing generalized spectral translation on function f adds a new term $x_{k_1}x_{k_2} \dots x_{k_{n/2}}$ where

$$k_1 \in \{i_1, j_1\},\ k_2 \in \{i_2, j_2\}, \dots, k_{n/2} \in \{i_{n/2}, j_{n/2}\}.$$

If the starting function is f_1 is and if $k_1 = 1$, $k_2 = 3$, and $k_3 = 6$, resulting function f_2 is:

$$f_2(x_1, x_2, x_3, x_4, x_5, x_6) = x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \oplus x_1x_3x_6$$

IV. Oscar-Bent functions

The bent function which does not have linear and constant members can be called Oscar-Bent function (the name derives from Oscar Rothaus, who first defined bent functions). For the bent function defined in (1), we can derive the Oscar-Bent function shown in (2) by using invariant spectral operations.

$$f_1(x_1, \dots, x_6) = x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \oplus x_1 \oplus 1 \qquad (1)$$

$$f_2(x_1, \dots, x_6) = x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \qquad (2)$$

To transform a bent function to its Oscar-Bent function we need to remove linear and constant members, which is done by using two invariant spectral operations: function complement and spectral translation. By counting only Oscar-Bent functions, we can deduce that the number of unique bent functions found with this algorithm is calculated by multiplying the number of Oscar-Bent functions with 2^{n+1}. The multiplier is found by calculating all possible combinations using two invariant operations mentioned above.

V. Genetic Algorithm

Genetic algorithm is a subclass of Evolutionary algorithm (EA), which is a subclass of Evolutionary computation and belongs to set of general stochastic search algorithm [10].

Population in both Genetic algorithms and in nature represents the set of individuals who are trying to survive and pass on their genes to the next generations. An individual can be interpreted as a set of genes and abilities, and how fit they are to survive in the current population and habitat. If we observe an individual as a solution to a problem, as well as in nature, optimization (survival of the fittest) will transpire, in the end, the fittest individual will represent an optimized solution to the problem. Given a population of individuals within some environment that has limited resources, competition for those resources causes natural selection (survival of the fittest). This in turn causes a rise in the fitness of the population. Given a quality function to be maximized, we can randomly create a set of candidate solutions, i.e., elements of the function's domain. We then apply the quality function to these as an abstract fitness measure – the higher the better. Based on these fitness values some of the better candidates are chosen to seed the next generation. This is done by applying recombination and/or mutation to them. Recombination is an operator that is applied to two or more selected candidates (the so-called parents), producing one or more new candidates (the children). The mutation is applied to one candidate and results in one new candidate. Therefore, executing the operations of recombination and mutation on the parents leads to the creation of a set of new candidates (the offspring). These have their fitness evaluated and then compete – based on their fitness (and possibly age) – with the old ones for a place in the next generation. This process can be iterated until a candidate with sufficient quality (a solution) is found or a previously set computational limit is reached [11].

The genetic algorithm can be described by the pseudo-code in Fig. 1.

```
InitializePopulation();
EvaluatePopulation();
while i < MaxIteration and
BestFitness < MaxFitness do
        Fitness = FitnessCalculation();
        Selection();
        ParentSelection();
        Reproduction();
        i++;
        BestFitness = Max(Fitness);
end while
return BestFitness
```

Fig. 1. Pseudo code detailing the genetic algorithm

A. Parent selection

Parent selection represents a strategy of selecting good parents to get a better next generation. The strategy should consist of some random chance in selection, so diverse parents will be used, and we can diverge from the local maximum (which can be reached by using the same group of parents).

- There are different strategies, for this paper, we have used:
- Roulette selection – odds of selection are determined by individual fitness and a corresponding piece of the roulette wheel is given; a random number is generated to represent a ball spin.
- Rang selection – is like Roulette selection, but fitness is scaled to give more chances to weaker individuals.
- Tournament selection – from a randomly selected group of individuals the best individual is chosen based on fitness.

B. Recombination and mutation

Recombination and mutation are used to produce a new solution to find the best one which solves the problem. Both methods may and may not be performed (based on chance which is determined on startup).

There are several recombination methods, but the most common is a crossover with one crossover point which is randomly selected. Genes from the first parent are copied to the crossover point, after which genes from the second parent are copied.

Mutation, if performed, results in randomly changing individual genes. For each gene, independently, it is determined whether the mutation will be performed or not.

C. Adult selection

Adult selection defines how will the new generation join the existing group of adults. Since the “habitat” can only sustain an already defined number of individuals, adult selection is needed to determine who will survive. Several methods are implemented:

- Full generational replacement – as the name applies, the parental generation is replaced with the new generation.
- Generational mixing – both generations are mixed, and the best of mixed generations survives.
- Overproduction – this method is a mixture between full generational replacement and generational mixing, in which the new generation has twice as many individuals as the parental generation, and only a half of the best child individuals survive to form the new parental generation.

Elitism can also be used, elitism enables keeping the best solution for the next solution, regardless of chosen adult selection.

VI. Genetic Algorithm For Bent Functions Generating

The problem can be defined by finding the most bent function by using the least number of invariant spectral operations. Since invariant spectral operations are performed on a bent function, a starting bent function needs to be defined. In our case, we start from the disjoint quadratic function.

For each implementation of a genetic algorithm, it is crucial to define an individual (which represents a solution to a problem) and a fitness function (which represents how good the solution is).

A. Individual representation

An individual is represented as a sequence of invariant operations which are performed on the most recent bent function.

B. Fitness function

It is recommended that the fitness function should be defined so it would have a minimum (the worst solution) and the maximum (the best solution), even though the boundaries can be arbitrary, the custom is to choose boundaries as 0 and 1.

To determine how good is the solution, we need to go back to the problem definition which states that we should find the most unique bent functions for the least number of invariant spectral operations. From this, we can derive that the fitness function can be calculated as the number of unique Oscar-Bent functions divided by the number of used invariant spectral operations.

By searching for the unique Oscar-Bent functions, we can generate the most bent functions, given that from the one Oscar-Bent function we can derive 2^{n+1} bent functions.

VII. Experimental Results

The application was developed in C#, and tests were performed on the laptop with the following configuration:

- CPU: Intel® Core™ i5-8250U CPU @ 1.6GHz
- RAM: 16 GB
- OS: 64bit Windows 10

Multiple parameters can be changed, and which can influence results (both performance and result wise). Testing all permutations of the possible combination of parameters is not a trivial task, and it is time-consuming. Therefore, some parameters were hardcoded with values that we perceived as best with our experience and using educated guesses.

Parameters that were hardcoded for all tests:

- Adult selection – Generational mixing

- Parent selection – Tournament selection
    - Tournament size – 20% of the population
- Possibility of gene mutation – 10%
- Possibility of recombination – 90%

A. Test 1 – Different number of genes

In this test we have chosen the number of variables to equal 6, population size is set to 10, and the number of generations is limited to 100. In this test, we will change the number of genes and compare the number of unique Oscar-Bent functions in an average of 5 runs. Results are shown in Table I.

TABLE I.
RESULTS OF TEST 1

Number of genes	**Number of generated OBF**	**Time (s)**
100	97.8	0.066
1 000	948.4	0.454
10 000	9 364.4	8.31
100 000	92 058	82.724

Through a different number of genes, we have seen that with linear growth of the number of genes, the number of unique Oscar-Bent functions grows in a linear fashion, with the growth factor between 9.5 and 10. When we analyze the time needed, it grows exponentially which is expected since the solution space grows exponentially as well.

B. Test 2 – Different number of variables

As in the previous test, the population size is set to 100, the number of generations is limited to 100 and the number of genes to 10 000. Here we will fluctuate number of variables. Results are shown in Table II.

TABLE II.
RESULTS OF TEST 2

n	**Number of generated OBF**	***Time (s)***
8	9 652.6	16
10	9 786.8	35.316
12	9 834.4	111.346
14	9 881.2	413.6

While an increasing number of variables we can observe that the number of unique Oscar-Bent functions increase with low percentages. Factor of growth for the time needed increases with each step, but it does not increase exponentially.

C. Test 3 – Application limits

In this test, we emphasized the performance limits of the application, not to the numbers we have, therefore we have run this test only once. Here, we have kept the number of genes to 10 000 and the number of generations to 100, as in the last test. But we have changed population size to 100. The results are shown in Table III.

TABLE III.
RESULTS OF TEST 3

n	**Number of generated OBF**	**Time (s)**
8	9 673	169.18
10	9 768	405.08s
12	9 845	1318.69s
14	9 877	1801.56s
16	N/A	N/A

VIII. CONCLUSION

We have seen that the usage of Genetic algorithm can be used in the generation of new bent functions. The performance of this approach indicates that future work can give promising results.

Memory is the biggest obstacle when working with bent functions. This problem can be approached by tracking only Oscar-Bent functions, which is performed in this paper. Further, all functions are kept in memory, which is a problem when expecting many unique Oscar-Bent functions, which we have seen in test 3. Future work will address this problem.

REFERENCES

[1] O. S. Rothaus, “On “Bent” Functions”, Journal of Combinatorial Theory, Series A, Vol. 20. No. 3, pp. 300-305, 1976.

[2] H. Dobbertin, “Construction of bent functions and balanced Boolean functions with high nonlinearity”, LCNS, vol. 1008, pp. 61-74, Springer, Berlin, Germany, 1995.

[3] J. Climent, F. Garcia, V. Requena, “On the iterative construction of bent functions”, in Proc. of the 5th WSEAS Int. Conf. on Inf. Security and Privacy (ISP06), pp. 15–18 World Scientific and Engineering Academy and Society (WSEAS), Wisconsin, USA, 2006

[4] J. Climent, F. Garca, V. Requena, “On the construction of bent functions of n+2 variables from bent functions of n variables”, Advances in Mathematics of Communications, vol. 2, pp. 421–431, 2008.

[5] C. Carlet, “A larger class of cryptographic Boolean functions via a study of the Maiorana McFarland construction”, LNCS, vol. 2442. pp. 549-564, Springer, 2002.

[6] H. Dobbertin, G. Leander, A. Canteaut, C. Carlet, P. Felke, P. Gaborit, “Construction of bent functions via niho power functions”, Journal of Combinatorial Theory, vol. 113, no. 5, pp. 779–798, 2006.

[7] M. Stanković, C. Moraga, R. Stanković, “Some Invariant spectral Operations for Functions with Disjoint Products in the Polynomial Form”, in Proc. EUROCAST 2017, LNCS, Vol. 10672, pp. 262-269, Springer, 2017.

[8] S. Stojković, M. Stanković, C. Moraga, R. Stanković, “Generation of Binary Bent Functions by Walsh Invariant spectral Operations Performed in Reed-Muller Domain”, Proceedings of 13th International Workshop on Boolean Problems, Bremen, Germany, pp. 255-266, September 19-21. 2018.

[9] R. S. Stanković, M. Stanković, C. Moraga and J. T. Astola, "Construction of Ternary Bent Functions by FFT-like Permutation Algorithms", 2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL), pp. 88-93, 2020.

[10] P. A. Vikhar, "Evolutionary algorithms: A critical review and its future prospects," 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, India, pp. 261-265, 22-24 Dec. 2016.

[11] A.E. Eiben, J.E. Smith, “Introduction to Evolutionary Computing”, 2nd ed., Berlin, Germany: Springer, 2015.

Application of Machine Learning Algorithms for Calculating Air Quality Index

Nebojša Bogdanović, Mladen Koprivica, *Member, IEEE*, Goran Marković, *Member, IEEE*

***Abstract*—Air pollution is an ever-growing issue, especially severe in urban and industrial areas. Air Quality Index (AQI) is a unit of measuring the level of air pollution, which takes into account the concentrations of all relevant air pollutants. There are two main problems that must be addressed in AQI calculations, i.e. regression and classification. The regression problem consists of calculating (approximating) the AQI index based on the concentrations of different air pollutants. In classification problem, the measurements of air pollutants' concentrations are classified into different Air Quality Classes. In this paper a number of Machine Learning (ML) and Deep Learning (DL) algorithms were designed and used in order to solve both the regression and classification problems for AQI. The main goal was to present performance comparison for wide set of ML and DL algorithms based on the values of Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Coefficient of Determination (R squared) in regression tasks, and Accuracy in classification tasks. Also, the percentage of algorithms' convergence and the time needed to perform these regression and classification tasks are also measured.**

***Index Terms*—Air Quality Index (AQI); Machine Learning; Deep Learning; Regression; Classification**

I. Introduction

Air pollution presents growing issue, which is especially severe in urban and industrial areas, and occurs whenever excessive quantities of pollutants such as gases, particulates, and bio-molecules are introduced into the atmosphere. It has harmful consequences on human population and other living organisms (i.e. it can cause diseases and/or even death, and impairing crops). Air pollutants can be solid particles, liquid droplets, or gases, and are classified as primary (i.e. directly emitted from the source) or secondary (i.e. formed in the atmosphere) pollutants. National environmental agencies set the standards and air quality guidelines regarding acceptable levels for air pollutants, while the air quality index (AQI) is used as an indicator in order to report the measuring of the air pollution and how unhealthy the air is (i.e. reports on possible associated health effects, above all for risk groups). AQI is calculated based on the maximum individual AQI measured for the observed criteria (air) pollutants, and this calculation is rather complex and thus its implementation is not suitable for applications with low-cost sensor platforms employed in the form of dense IoT-based sensor network. In fact the process of calculating AQI by formulas consists of two steps: (1) calculation of air quality index for every pollutant in each of the measurements separately, and (2) observing values of the all indexes for every measurement in order to find the maximum. On the other hand, in a case of application of machine learning (ML), a whole process of training and testing the algorithms takes longer than the use of formulas, but these algorithms only need to be trained once, before its application in real-time systems. Thus, when compared to formulas which needs to be used every time we have a different measurement, the time needed to perform this task by ML algorithms is shorter. It should be noticed, that formulas used for these calculations are not complex, but since the implementation of these formulas requires using multiple loops and case functions, the process takes longer when compared to the testing part of the ML and DL algorithms.

On the other hand, the more useful, flexible and scalable usage of AQI in terms of influence on the human population health, would be to deploy air quality forecasting system based on the measured levels of concentration of individual air pollutants, which would be able to predict AQI (i.e. air quality) locally and in short-term manner (hourly). This demands the use of dense network of low-cost sensors and thus requires simple solution for the determination of AQI based on the local low-quality air pollutant measurements.

So far, research community and environmental agencies have developed different methods for calculation of AQI, [1][2], but still no universally accepted method exist that is appropriate in all scenarios, [3]. The machine learning (ML) based methods are proposed as an obvious and natural solution for AQI determination and prediction, such as fuzzy lattices decision support system, [1], the support vector regression (SVR), [2], or different ML algorithms (linear regression, random forest, decision tree, SVR, and K-Nearest neighbor).

In this paper, the broad set of ML algorithms, including deep learning (DL), are observed as possible solutions for determination of AQI based on the measured levels of six criteria pollutants. Also, we here addressed two main issues in AQI calculation: regression problem that represents AQI calculation based on criteria pollutants concentrations, and classification problem in which the measurements of air pollutants' concentrations are classified into the Air Quality Classes. The output of the ML models is the approximation of the current values of AQI, while the prediction of the future values of AQI is something we are considering for the future works. In total, 8 different ML algorithms and 5 DL models were analyzed for the regression task, while 9 different ML algorithms and 3 DL models were observed for the classification task. We here observed much broader set of ML algorithms than in previous work, i.e. in [3]. ML algorithms and DL models were designed, optimized and tested based on dataset consisting of real-time measurements

Nebojša Bogdanović is a student at the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11020 Belgrade, Serbia (e-mail: bn150118d@student.etf.bg.ac.rs).

Mladen Koprivica is with the University of Belgrade, School of Electrical Engineering, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia (e-mail: kopra@etf.bg.ac.rs).

Goran Marković is with the University of Belgrade, School of Electrical Engineering, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia (e-mail: gmarkovic@etf.bg.ac.rs).

gathered from 5 countries. The performance metrics are defined, and performance analysis and comparison of the observed ML algorithms and DL models is performed for regression and classification tasks.

The paper is organized as follows. In the section II the basic concepts related to air quality pollutants, monitoring, and scale are given. Section III gives short description of the observed machine learning algorithms and the deep learning models observed in this paper, as well as a short description of AQI regression and classification tasks, while a dataset used in ML and DL algorithm training and performance analysis is described in section IV. The main results and conclusion are presented in section IV, followed by the final concluding remarks.

II. Air Pollutants and Air Quality Scale

Air pollution is most frequently man-made. It usually comes from factories, powerplants and heating plants which use unrenewable energy sources, cars and public transport.

According to International Energy Agency (IEA) [4], from the year 2018 to 2040 the projected energy demand should rise annually by 1.3%. This projected growth can be seen on Fig. 1, where in the year 2020 around 60% of all the energy should be generated using non-renewable energy sources. While the use of renewable energy sources is projected to increase by the year 2040, because of the growth in energy demand, the amount of energy generated by coal, gas, oil and nuclear energy will not decrease.

This is a growing problem, in both the developed and in countries in development, because a usage of fossil fuels results in high concentrations of air pollutants released into the atmosphere. Many of developed countries are fighting this problem by imposing laws which are restricting the amounts of fossil fuels burned each year. Also, powerplants and factories are required to use filters in order to reduce the emission of air pollution into the atmosphere.

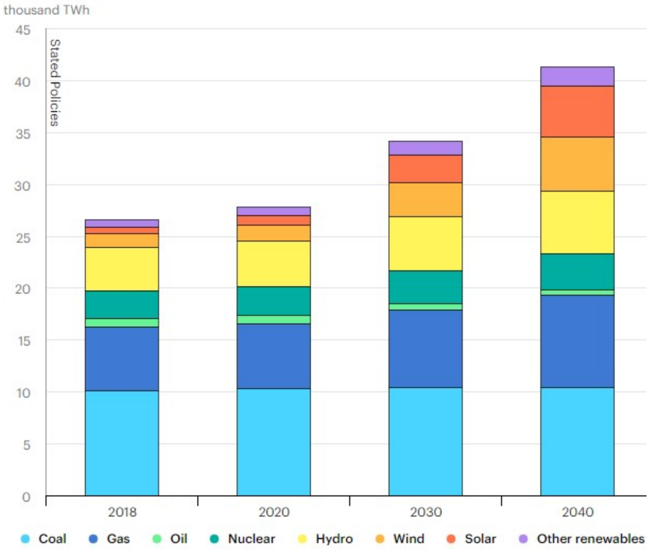


Fig. 1. Projected growth of energy demand from 2018 to 2040, [4]

The most common types of air pollution according to New South Wales Ministry of Health (NSW Health) [5], are listed below:

- Carbon Monoxide (CO), mostly generated by motor vehicles and industry plants;
- Ozone (O_3), the main component of smog, a product of interaction between sunlight and emissions from motor vehicles and industry plants;
- Particulate Matter (PM 2.5, PM 10), the small solid particles and liquid droplets suspended in air, made up of variety of components including nitrates, sulfates, organic chemicals, metals, soil or dust particles and allergens, which mostly comes from motor vehicles and industry plants;
- Nitrogen Dioxide (NO_2), generated by motor vehicles, industry plants, and unflued gas-heaters; and
- Sulphur Dioxide (SO_2), generated by the fossil fuel combustion at power plants and industrial facilities.

A. *Air Quality Scale*

Air Quality Index (AQI) can be calculated in a number of different ways, and depending on which formulas are used for calculations, there are different AQI scales. In this paper, the formulas and scales used are created by the Central Pollution Control Board, Ministry of Environment, Forests and Climate Change in India. This corresponding air quality scale is presented in Table I.

TABLE I
Air Quality Scale (India)

Category	AQI Index	Possible Health Impacts
Good	0-50	Minimal health impacts
Satisfactory	51-100	Minor breathing discomfort to sensitive people
Moderate	101-200	Breathing discomfort to the people with lung, asthma and heart diseases
Poor	201-300	Breathing discomfort to most people on prolonged exposure
Very Poor	301-400	Respiratory illness on prolonged exposure
Severe	401-	Affects healthy people and seriously impacts those with existing diseases

III. Algorithms and Problems

As defined in the introduction, there are two types of challenges in calculating AQI index which are addressed in this paper, the regression and the classification tasks. Both of these issues hold valuable information when calculating levels of air pollution. Some of the reasons for using ML algorithms in this area are:

- Provision of real-time decision support for air quality sensors, especially in a case of wide usage of low-cost sensors (i.e. for IoT-based environmental monitoring networks). Specifically, a verification that sensors for monitoring concentrations of various pollutants are working well, to predict the missing values in a case of sensor malfunction, and to evaluate inputs and decide whether and alarm should be triggered or not [1];
- Improvement of sensor performance for lower-cost air quality monitoring [6]; and
- Forecasting (prediction) of future values of pollution concentrations and AQI index [2] [3].

In this paper, we have performed comparison of the wide set of various machine learning algorithms for regression and classification tasks, such as: Multiple Linear Regression (MLR), Stochastic Gradient Descent (SGD) Classifier based on Linear Regression, Support Vector Machine (SVM), K-

Nearest Neighbors (KNN), Random Forest (RF), Decision Tree, Extra Trees Regression, Adaptive Boosting based on Decision Trees (AdaBoost), and Gradient Tree Boosting (GradBoost). Also, we designed and estimated performance for several Deep Learning algorithms in both tasks

Regression and classification tasks are rather similar, and thus the classification task can be realized by classifying the results achieved by regression algorithms into the respective categories in Table I. In this case, we would achieve 100% accuracy for classification task for the all algorithms, except for Multiple Linear Regression, but a time needed to execute would be even higher than for the regression algorithms. This is why in this paper we proposed a different method. In our method, the input data used for classification algorithms is the same as for the regression algorithms, and that is just the concentrations of the pollutants of measurements, while the labels used for training of the ML algorithms and DL models are final categories in Table I. By using this method we expected slightly lower classification accuracy (which will be discussed in section VI), when compared to the first method (based on regression), but the time needed to execute such algorithms would be, depending on an algorithm, from two to ten times smaller than for their respective regression algorithms.

The performance metrics for AQI regression algorithms were the values of Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2), while for the classification tasks we used the algorithm accuracy as the main performance metric. As the additional performance metrics for both tasks, we measured percentage of convergence and the elapsed time needed to perform these tasks for all observed algorithms.

IV. Air Pollutant Measurement Dataset

The dataset used in the analysis was created from data gathered from websites data.world [7], and openaq.org [8], and it consists of 35440 independent measurements from 5 countries (Serbia, India, USA, Australia and Taiwan). The measurements data were gathered from 2016 to April 2021. Each measurement consists of measured concentrations of Carbon-monoxide (CO), Ozone (O_3), particulate matter PM 2.5 and PM 10, Nitrogen-dioxide (NO_2) and Sulphur-dioxide (SO_2). The concentrations of all of the pollutants are measured in $\mu g/m^3$, except for CO, which is measured in mg/m^3. The mean values (mean) and standard deviations (std) of the measurements in dataset, are given in Table II.

TABLE II
Dataset description

	Count	Mean	Std
CO	35440	1.39088	1.34341
O_3	35440	51.9751	61.5998
PM 2.5	35440	76.7299	112.159
PM 10	35440	152.993	185.914
NO_2	35440	51.9291	56.3290
SO_2	35440	8.51487	8.71780
AQI	35440	202.835	195.820

Based on these concentrations, the reference AQI index and the air quality class were calculated for each of the measurements, by using formulas implemented in Python scripts. The dataset was divided into training and test sets in the ratio 90%:10%, and the training set was further divided into training and validation sets in the same ratio.

V. Results of Performance Analysis

The performance analysis of observed machine learning algorithms is performed by using Scikit-learn library for the Python programming language, while the deep learning algorithms were implemented using Tensorflow and Keras libraries for Python. The implementations were executed on Google Colaboratory cloud computing platform, by using Intel(R) Xeon(R) CPU @ 2.30 GHz processing unit with 16 GB of available RAM memory.

Both the machine learning and deep learning algorithms were trained and tested independently 42 times, with the values for *random_state* parameter ranging from 0 to 41, in order to guarantee different train/test splits of the dataset for the each iteration of the observed algorithm.

A. Regression algorithms

The analysis showed that all of the regression algorithms have the convergence rate of 90.47% (38/42). In 4 executions where the algorithms diverge, the corresponding MAE and RMSE values were not taken into account in the calculation of the mean values and standard deviations of these errors.

The five Deep Learning models, marked DL#1 to DL#5, were designed, optimized and used. These neural networks models are defined as: DL#1 model with 3 hidden layers comprising of with 128 neurons in each layer, DL#2 model with 3 hidden layers with 256 neurons in each layer, DL#3 model with 3 hidden layers with 512 neurons in each layer, DL#4 model with 3 hidden layers with 128 neurons in the first hidden layer, 1024 neurons in the second hidden layer, and 128 neurons in the third hidden layer (DL#4), and DL#5 model with 3 hidden layers with 256 neurons in the first hidden layer, 1024 neurons in the second hidden layer, and 256 neurons in the third hidden layer. Data is normalized in input layer of each DL model. The activation function for all layers was a ReLU function, while the loss function used was Mean Squared Error (MSE). The Adam optimization function was used with the learning rate of 0.001, and every neural network model is trained over 100 epochs. Different numbers of epochs for training DL models were considered during the design of these models, for both regression and classification tasks. In this process, it is observed that even if for some of the lower numbers of epochs the algorithms performed similarly as for 100 epochs, the results were not consistent enough, e.g. the convergence rate was lower (i.e. for 80 epochs the algorithms converged in 28/42 cases). Thus, we choose the number of 100 epochs, since the further rise in the number of epochs did not give better results.

The mean values and standard deviations of MAE and RMSE for all of the algorithms used in AQI regression task are shown in Table III, while in Table IV the times needed to train (single execution) of all these algorithms are given. The time needed for execution of all trained algorithms for one test measurement was similar and very short (in ms).

From the MAE and RMSE values, shown in Table III, it can be inferred that the best overall performance in a case of regression was achieved by using Adaptive Boosting based on Decision Trees. Also, by analyzing tables III and IV it can be inferred that the simpler algorithms, such as Multiple Linear Regression and Decision Tree take shortest time to train (and execute). On the other hand, the algorithms that consist of a large number of decision trees (i.e. Random Forest, AdaBoost or Extra Trees), SVM and deep learning algorithms, take the longest time to train (and execute) due to the complexity.

TABLE III
REGRESSION ALGORITHMS - MAE AND RMSE VALUES

Algorithm	MAE		RMSE	
	Mean	Std	Mean	Std
Random Forest	0.515066	0.065481	4.202714	1.167030
Decision Tree	0.652400	0.094673	5.911044	1.294427
AdaBoost	0.102419	0.02371	1.056525	0.477244
GradBoost	0.492849	0.049863	3.154737	0.952986
Extra Trees	0.469917	0.043560	2.800131	0.763884
KNN	7.586408	0.243762	17.655297	1.181481
SVM	6.811546	0.204737	12.891746	1.510237
MLR	35.95306	0.51608	52.938264	1.547197
DL#1	1.857936	0.407002	3.732621	0.413279
DL#2	1.552141	0.340354	3.413873	0.465218
DL#3	1.819313	0.501331	3.687509	0.576523
DL#4	1.616337	0.356425	3.417748	0.458047
DL#5	1.719751	0.539735	3.565049	0.655192

TABLE IV
REGRESSION ALGORITHMS - DURATION OF TRAINING (SINGLE EXECUTION)

Algorithm	Time [s]		Alg.	Time [s]	
	mean	std		mean	std
Random Forest	142.762	8.4116	MLR	0.007	0.0112
Decision Tree	0.215	0.0043	DL#1	170.780	27.4982
AdaBoost	69.558	2.3812	DL#2	250.169	16.4083
GradBoost	42.799	1.1223	DL#3	494.753	20.3062
Extra Trees	103.272	2.2578	DL#4	354.726	24.7223
KNN	0.499	0.0156	DL#5	627.936	30.0591

Furthermore, a more detailed statistical and error analysis (i.e. the minimum and maximum error values, the threshold values corresponding to 25%, 50% and 75% of instances), as well as time needed for training of AdaBoost algorithm are shown in Table V.

TABLE V
DETAILED ANALYSIS OF ADABOOST ALGORITHM FOR REGRESSION TASK

	MAE	MSE	RMSE	R^2	Time [s]
Mean	0.10242	1.338014	1.056525	0.999965	69.558
Std	0.02371	1.122726	0.477244	0.000029	2.3812
Min	0.06659	0.178894	0.422959	0.999895	64.468
25%	0.08444	0.40364	0.635299	0.999943	67.793
50%	0.09975	0.807562	0.897779	0.999978	69.868
75%	0.11428	2.176002	1.474417	0.999989	71.291
Max	0.15632	3.988713	1.997176	0.999995	75.529

As obvious in Table V, 50% of the MAE values for AdaBoost algorithm are under 0.1, with its mean value being just over 0.1. These are by far the best values of MAE for all of the observed regression algorithms that were compared in this paper.

B. Classification algorithms

The analysis showed that all the observed classification algorithms have the convergence rate of 100%, which means that the algorithms manage to converge around the mean values of accuracy in all of the 42 independent executions. Besides machine learning algorithms, three Deep Learning models, marked DL#6 to DL#8, were designed, optimized and used. These neural networks models are defined as: DL#6 model with 2 hidden layers with 128 neurons in each layer, DL#7 model with 2 hidden layers with 256 neurons in each layer, and DL#8 model with 2 hidden layers with 512 neurons in each layer. Data is normalized in the input layer of every neural network model. The activation function of each hidden layer is the ReLU function and the activation function of the output layer is the Softmax function. The loss function is Binary Cross-entropy, and the metrics of the loss function is the binary accuracy function with the 0.5 threshold value. The Adam optimization function was used with the learning rate of 0.001, and every neural network model is trained over 100 epochs.

The mean values and corresponding standard deviations (std) of classification accuracy for all observed classification algorithms, as well as the times needed for the training are given in Table VI.

Based on accuracy values for different algorithms, shown in Table VI, it can be inferred that the best algorithm for the classification task is the Gradient Tree Boosting. Also, the difference in time needed to train (and execute) more and less complex classification ML algorithms is not as big as it is in case of regression algorithms. This can be explained by the fact that the classification problem is easier to solve, and it does not require as much time as the regression one. Deep learning algorithms for classification take longer to train, since these were trained over 100 epochs.

TABLE VI
CLASSIFICATION ALGORITHMS - ACCURACY AND DURATION OF TRAINING

Algorithm	Accuracy		Time [s]	
	mean	std	mean	std
Random Forest	0.998683	0.00052	26.04913	0.303142
Random Forest Hybrid	0.998388	0.000612	18.30846	0.115094
Decision Tree	0.99822	0.000775	0.096139	0.003632
AdaBoost	0.998233	0.000753	0.104626	0.003506
GradBoost	0.999422	0.000432	24.62098	3.891909
Extra Trees	0.990682	0.001651	12.96546	0.247514
KNN	0.92313	0.004718	0.232663	0.008677
SVM	0.976849	0.00237	67.03444	6.434671
SGD	0.695058	0.009976	0.444438	0.019513
DL#6	0.994421	0.001118	118.5808	22.9412
DL#7	0.994591	0.001186	139.3796	2.512952
DL#8	0.994536	0.001218	440.6514	24.03057

The more detailed analysis of the Gradient Tree Boosting algorithm is shown in Table VII (the minimum and the maximum accuracy values are given, as well as threshold values corresponding to 25%, 50% and 75% of instances, and time needed for training), while estimated confusion matrix for this algorithm is shown on Fig. 2. It can be seen that in a case of Gradient Tree Boosting algorithm, only 3 of 7088 independent measurements of the test set used were misclassified (see confusion matrix in Fig. 2).

When compared to the classification results achieved in [3], we here achieved slightly better results for classification accuracy for the same algorithms that were used in both papers. However, in this paper the number of epochs for training the DL algorithms was higher than in [3], which can be one of the reasons for better accuracy results. Yet, the novelty of our paper, when compared to the work in [3], is that we included a number of algorithms that were not implemented in [3], for which we here achieved even better results in classification accuracy.

TABLE VII
DETAILED ANALYSIS OF GRADBOOST ALGORITHM

	Accuracy	Time [s]
mean	0.999422	24.62098
std	0.000432	3.891909
min	0.998025	22.1305
25%	0.999154	22.70442
50%	0.999436	22.99523
75%	0.999718	23.20253
max	1	35.08803

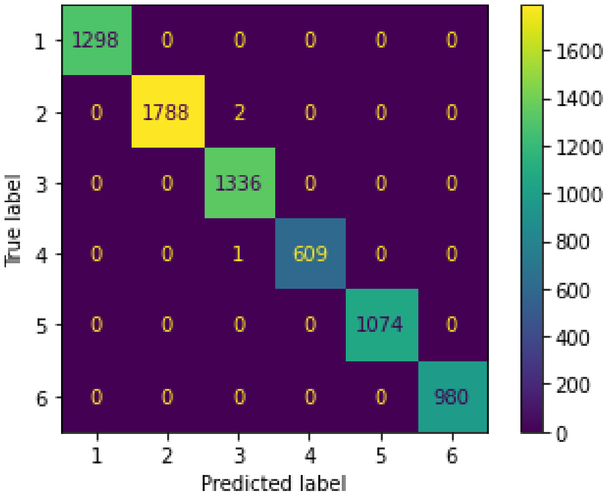


Fig. 2. Confusion matrix for the GradBoost algorithm

VI. CONCLUSION

Alongside global warming, the air pollution is one of the most alarming global ecological problems. Thus, developed countries, international health organizations, as well as some international companies are investing money in to reduce the impact air pollution have on global health. Also, some of air pollution aware companies try to motivate people to contribute to the cause, by giving them a chance to connect their air quality sensors with the global network of sensors, created by these companies. I.e., one of the most famous companies and websites that does this is called IQ Air [8].

The main topics covered in this paper are calculating the AQI (regression task), and the classification of air pollutant measurements into different air quality classes. We observed a wide set of machine learning and deep learning regression and classification algorithms for these tasks, and presented the performance comparison of these algorithms, based on the values of MAE, RMSE and accuracy, as well as the time needed to execute these algorithms. In total, 8 and 9 ML algorithms, as well as 5 and 3 DL models, were observed for regression and classification tasks, respectfully. It is shown that the AdaBoost algorithm presents best choice in the case of regression task, while the GradBoost algorithm presents the best choice in the case of classification tasks.

The presented results, as well as the designed and trained algorithms, present a foundation of a forecasting model, for predicting the missing and future pollution measurements and values of air quality index. This forecasting model could be used as a part of mobile application, which would inform users about the daily and weekly predictions of the pollution levels. This is one of the ideas for the future works. Another possible way of using the designed and trained algorithms, would be implementing in industrial plants.

ACKNOWLEDGMENTS

This work has been partly supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

REFERENCES

[1] I. N. Athanasiadis, V. G. Kaburlasos, P. A. Mitkas, and V. Petridis, “Applying Machine Learning Techniques on Air Quality Data for Real Time Decision Support”, 1st Intl. Symposium on Information Technologies in Environmental Engineering (ITEE 2003), pp. 51, 2003, ICSC/NAISO Academic Press.

[2] M. Castelli, F. M. Clemente, A. Popovič, S. Silva, and L Vanneschi, "A Machine Learning Approach to Predict Air Quality in California", Complexity, vol. 2020, Article ID 8049504, 23 pages, 2020.

[3] M. Sharma, J. Samyak, S. Mittal, and T. Sheakh, “Forecasting and Prediction of Air Pollutants Concentrates Using Machine Learning Techniques: The Case of India”, IOP Conference Series: Materials Science and Engineering, ICCRDA 2020, Vol. 1022, 012123, 2021.

[4] Electricity generation by fuel and scenario, 2018-2040, IEA, Paris https://www.iea.org/dataand-statistics/charts/electricity-generation-by-fuel-and-scenario-2018-2040 (last time accessed on 10.06.2021.)

[5] https://www.health.nsw.gov.au/environment/air/Pages/common-air-pollutants.aspx (last time accessed on 10.06.2021.)

[6] N. Zimmerman, A. A. Presto, S. P. N. Kumar, J. Gu, A. Hauryliuk, E. S. Robinson, A. L. Robinson and R. Subramanian, "A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring", Atmos. Meas. Tech., vol. 11, no. 1, 2018, pp. 291–313.

[7] https://data.world/ (last time accessed on 10.06.2021.)

[8] https://openaq.org/ (last time accessed on 10.06.2021.)