ALGORITMA OPTIMASI CHAOS PADA RIDGE POLYNOMIAL NEURAL NETWORK UNTUK KRIPTANALISIS KUNCI PUBLIK ELGAMAL

Rina Pramitasari

Teknik Komputer Universitas AMIKOM Yogyakarta
email : rina.pramitasari@amikom.ac.id

Abstraksi

Kriptografi kunci publik ElGamal adalah kriptografi yang berdasarkan pada permasalahan logaritma diskrit. Tingkat kesulitan menemukan logaritma diskrit merupakan tingkat keamanan pada kriptografi kunci publiuk ElGamal. Kelemahan kriptografi kunci public ElGamal adalah membutuhkan akan keacakan dan kecepatan yang sangat lambat. Kelemahan lainnya adalah perluasan pesan dengan faktor dua yang terjadi selama enkripsi. Berdasarkan kelemahan tersebut diusulkan metode Algoritma Optimasi Chaos dan Ridge Polinomial Neural Network untuk kriptanalisis kunci public ElGamal. Nilai awal bobot Ridge Polynomial Neural Network menggunakan hasil dari Algoritma Optimasi Chaos karena untuk mengatasi iterasi yang terjebak ke dalam minimum local. Variabel chaos di dalam range tertentu memiliki beberapa fitur yaitu Randomness, property ergodis dan regularity. Fungsi chaos sederhana adalah Logistic Map. Hasil menunjukkan bahwa penelitian menghasilkan kunci privat dengan akurat dan baik dimana menghasilkan iterasi yang konvergen.

***Kata Kunci* :**
Algoritma Optimasi Chaos, Ridge Polynomial Neural Network, Kriptografi Kunci Publik ElGamal, Kriptanalisis.

Abstract

ElGamal's public key cryptography is cryptography based on the discrete logarithm problem. The difficulty level of finding discrete logarithms is a security level in ElGamal public key cryptography. The weakness of ElGamal public key cryptography is that it requires randomness and very slow speed. Another drawback is the expansion of the message by a factor of two that occurs during encryption. Based on the weakness of the proposed method, Chaos Optimization Algorithm and Ridge Polynomial Neural Network for ElGamal public key cryptanalysis. The initial value of the weights of the Ridge Polynomial Neural Network uses the results of the Chaos Optimization Algorithm because it overcomes iterations that are stuck into the local minimum. Chaotic variables within a certain range have several features, namely randomness, ergodic properties and regularity. A simple chaos function is a Logistics Map. The results show that research produces private keys accurately and well which results in convergent iterations.

***Keywords* :**
Chaos Optimization Algorithm, Ridge Polynomial Neural Network, ElGamal Public Key Cryptography, Cryptanalysis.

1. Pendahuluan

Algoritma ElGamal adalah kriptosistem kunci publik berdasarkan permasalahan logaritma diskrit. Algoritma tersebut terdiri dari algoritma enkripsi dan algoritma tanda tangan. Algoritma tanda tangan ElGamal mirip dengan algoritma enkripsi dimana kunci publik dan kunci privat memiliki bentuk yang sama. Namun, algoritma enkripsi tidak sama dengan algoritma tanda tangan verifikasi. Kelemahan utama Algoitma ElGamal adalah membutuhkan akan kerandoman, dan kecepatan yang lebih lambat terutama algoritma tanda tangan. Potensi kerugian lainnya dari algoritma ElGamal adalah perluasan pesan dengan faktor dua terjadi selama enkripsi. Namun, pesan seperti ekspansi dapat diabaikan jika kriptosistem hanya digunakan untuk pertukaran kunci rahasia enkripsi ElGamal digunakan dalam GNU privacy Guard Software gratis, PGP versi terbaru, dan kriptosistem ElGamal lainnya tidak secara semantic aman. Algoritme El-Gamal tidak hanya dapat digunakan dalam enkripsi data, tetapi juga dalam tanda tangan digital dan keamanannya bergantung pada masalah logaritma divergensi dalam domain terbatas [1][2].

Chaos adalah fenomena nonlinier umum, yang tindakannya adalah kompleks dan mirip dengan kekacauan. Definisi matematika sebagai perilaku semi-acak yang dihasilkan oleh sistem deterministik nonlinear. Secara umum, Chaos memiliki beberapa sifat dinamis yang penting, seperti ketergantungan sensitif pada kondisi awal, bersifat quasi-stokastik dan ergodisitas. Gagasan menggunakan sistem chaos sebagai proses secara acak telah diperhatikan di banyak bidang. Satu satunya pada bidang teori optimasi. Chaos memainkan peran penting dalam perhitungan optimasi. Chaos memiliki keacakan dan ergodisitas, yang membantu algoritma untuk melewati minimum local [5].

Pi-Sigma Neural Network (PSN) adalah pertama diperkenalkan oleh Shin dan Ghosh. Jaringan feedforward dengan satu lapisan hidden

dan unit perkalian lapisan output. PSN menghitung perkalian penjumlahan unit pada lapisan output dan menyebarkannya ke fungsi nonlinier. PSN dapat belajar dengan stabil walaupun dengan learning rate yang cukup besar. Penggunaan unit penjumlahan linier membuat analisis konvergensi pada aturan pembelajaran untuk PSN lebih akurat dan mudah ditelusuri. Ridge Polynomial Neural Networks (RPNN) diperkenalkan oleh Shin dan Ghosh. RPNN adalah generalisasi dari Pi-Sigma Neural Networks. RPNN dibangun dengan menambahkan derajat yang berbeda dari PSN sebagai bangunan dasar. Menggunakan polinomial univariat dan menyediakan struktur yang efisien dan teratur dibandingkan dengan jaringan feedforward tingkat tinggi biasa. RPNN dapat mendekati fungsi kontinu multivariat apapun pada himpunan kompak dalam ruang input multidimensi, dengan tingkat akurasi baik. Sama dengan jaringan saraf PSN, RPNN hanya memiliki satu lapisan bobot adaptif dan mempertahankan semua keunggulan PSN [4].

Penelitian ini membahas game teori, contohnya permainan zero-sum, ada banyak studi teoretis mutakhir bahwa perilaku rata-rata waktu para agen pada akhirnya akan mengarah pada konvergensi ke ekuilibrium Nash, yang menunjukkan stabilitas ekonomi sampai batas tertentu. Namun pada kenyataannya, dari perspektif perilaku, sangat tidak masuk akal hanya fokus pada perilaku agen rata-rata waktu, dan dalam penelitian ini, sehingga berpendapat bahwa dalam konteks permainan zero-sum, perilaku sehari-hari di online dinamika belajar sulit diprediksi. Secara khusus, algoritme pendakian dan penurunan gradien pada akhirnya akan menyebabkan kekacauan Lyapunov di ruang pembayaran. Penelitian ini mensimulasikan dan menganalisis algoritme yang terkait dengan fenomena chaos, dan menyangkal kesimpulan bahwa "perilaku rata-rata waktu dalam pembelajaran online pada akhirnya akan menyatu dengan kesetimbangan Nash" [11].

Penelitian ini membahas algoritme pencarian sparrow chaos baru yang dirancang untuk memecahkan masalah TSP. Pertama, metode greedy dan chaos kubik digabungkan untuk menginisialisasi populasi. Kedua, formula baru untuk pembaruan lokasi pengintaian dan peringatan dini. Kemudian, pencarian lokal dilakukan dengan menggabungkan mutasi Gaussian, mutasi penyisipan, dan operasi urutan terbalik untuk membuat algoritma melompat keluar dari optimum lokal. Dan dalam proses iterasi, kebugaran populasi dipantau, solusi optimal global historis diperkenalkan tepat waktu untuk mengoordinasikan kemampuan pencarian lokal dan global dari algoritma. Hasilnya menunjukkan bahwa algoritma pencarian novel chaos sparrow memiliki keuntungan yang jelas dalam menyelesaikan masalah TSP, baik dalam akurasi maupun efisiensi [12].

Penelitian ini menetapkan tingkat konvergensi untuk memasang jaringan saraf polinomial dengan fungsi aktivasi kuadrat melalui algoritma mini-batch Stochastic Gradient Descent (SGD). Secara khusus, kami fokus pada implementasi paralel penghitungan gradien mini-batch pada platform komputasi terdistribusi. Kami pertama-tama mengilustrasikan bahwa SGD konvergen pada laju linier ke solusi optimal, dan laju konvergensi dapat dicirikan sebagai fungsi dari ukuran mini-batch. Selanjutnya, kami menggunakan SGD dengan pendekatan terdistribusi di beberapa prosesor, di mana gradien mini-batch parsial dihitung dan dikuantisasi untuk dikirim ke prosesor master di setiap iterasi, menghasilkan algoritma Quantized Stochastic Gradient Descent (QSGD). QSGD terbukti mempertahankan tingkat konvergensi serupa dari SGD ke solusi optimal global sambil secara signifikan mengurangi biaya komunikasi [8].

Penelitian ini membahas permasalahan ketidakseimbangan kelas yang sering terjadi dalam diagnosis penyakit, yang secara signifikan mempengaruhi model diagnostik berbasis kecerdasan buatan dan menyebabkan biaya besar tentang kesalahan klasifikasi penyakit. Untuk meringankan masalah keseimbangan kelas maka mengusulkan algoritma class imbalance-oriented polynomial neural network (CIPNN), yang menggabungkan pengambilan sampel data dan klasifikasi ensemble. Secara khusus, kami menggunakan algoritma tetangga terdekat untuk mengidentifikasi area kritis dari yang diberikan dataset medis (dataset asli), yang membentuk dataset area kritis. Penelitian yang dilakukan pada sembilan set data medis yang tidak seimbang menunjukkan bahwa metode yang diusulkan dapat secara efektif mengurangi masalah ketidakseimbangan kelas [6]. Kemudian penelitian yang diusulkan adalah kriptanalisi kunci public ElGamal dengan menggunakan Algoritma Optimasi Chaos pada Ridge Polynominal Neural Network.

2. Metode Penelitian

Berikut langkah penelitian pada Gambar 1 berikut untuk mengkriptanalisis kunci public ElGamal dengan menggunakan Algoritma Optimasi Chaos pada Ridge Polynominal Neural Network.

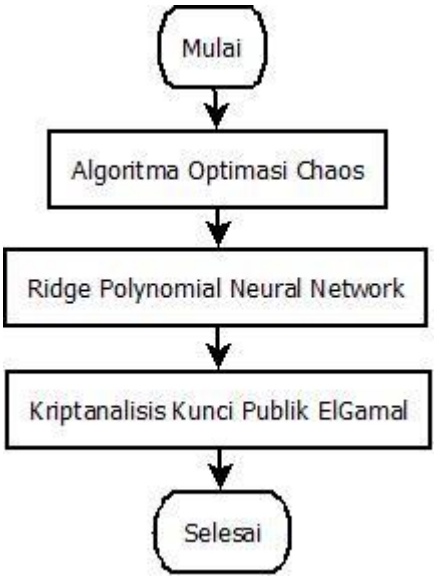
1). Tetapkan parameter awal pada jaringan syaraf adalah jumlah neuron pada PSN, jumlah maksimum iterasi = 5000, error yang diinginkan pada RPNN = 0.0001, error yang diinginkan pada PSN = 0.00018, learrning rate.

2). Semua data dinormalkan pada rentang nilai [0, 1]

3). Tetapkan parameter awal pada COA adalah p = 0.4, pencarian pertama [a0, b0] = [-50, 50], pencarian kedua [a1, b1] = [-3, 3] , maksimum iterasi pertama = 500 ; maksimum iterasi kedua = 800. Hasil dari COA menjadi nilai awal pada RPNN.

4). Pelatihan data pada kriptanalisis kunci public ElGamal menggunakan RPNN

5). Pengujian data pada kriptanalisis kunci public ElGamal menggunakan RPNN. Hasil yang baik dipenuhi berdasarkan pada nilai MSE.
6). Semua data hasil pengujian di denormalisasi [9][10].



Gambar 1 Langkah Penelitian

2.1. Algoritma ElGamal

a). Metode untuk menghasilkan pasangan kunci. Pertama, pilih bilangan prima besar p, dua bilangan random g dan s, g < p, s < p, dalam bidang terbatas CF(p). Hitunglah :

$$y = g \wedge s \ (mod\ p) \quad (1)$$

Kunci publik adalah y, g dan p. Kunci privat adalah s. Kemudian g dan p dapat dibagikan dalam grup.

b). Metode untuk proses enkripsi, mengatur informasi terenkripsi sebagai M.
Pertama, pilih secara random bilangan t, t dan *p – 1* adalah bilangan prima relatitif. Hitunglah :

$$a = g \wedge t \ (mod\ p) \quad (2)$$
$$b = y \wedge t\, M \ (mod\ p)$$

(a, b) adalah ciphertext, dua kali ukuran plaintext.

c). Metode untuk proses dekripsi

$$M = b \,/\, a \wedge s \ (mod\ p) \quad (3)$$

Ketika mengenkripsi, pengguna dapat memilih bilangan random *t* ($0 < t < p - 1$), jadi enkripsinya adalah karakteristik probabilitas. Khususnya, yang utama bilangan p harus cukup besar. Keamanan ElGamal terutama bergantung pada p dan g, dan kita harus memastikan bahwa g dan faktor prima besar dari p-1 tidak dapat direduksi [3][8][9].

2.2. Algoritma Optimasi Chaos

a). Pembangkit Chaotic Sequences

Biasanya varian chaotic dihasilkan oleh Logistic Map yang dijelaskan sebagai berikut [7][10][14].

$$\begin{cases} y_{n+1} = \mu y_n (1 - y) \\ 0 \le y_1 \le 1 \end{cases}, \quad n = 1, 2, \cdots \quad (4)$$

Perlu dicatat bahwa nilai seperti 0, 0,25, 0,5, 0,75 dan 1 adalah titik periodik. Dan nilai awal y1 dari varian chaos tidak bisa salah satunya. Ketika parameter kontrol diatur ke 4, sistem dalam keadaan kacau. Diberikan y1 nilai acak, urutan kekacauan yang ditentukan akan diperoleh dengan menggunakan (4).

b). Prinsip Dasar Algoritma Optimasi Chaos (COA)

COA menghitung solusi yang layak menggunakan karakteristik ergodisitas dari varian chaotic. Langkah COA ditunjukkan sebagai berikut:

Langkah 1) Menghasilkan urutan chaotic menggunakan persamaan evolusi chaotic. Perhatikan bahwa interval urutan chaotic antara 0 dan 1.

1). Inisialisasi
Tetapkan *r* = 0, *r max*, $A_i^0 = \in (0,1)$, dimana *i* = 1, 2, ..., *n* adalah jumlah bobot diantara layer input, layer hidden dan bobot bias.
Pastikan $A_i^0 \neq (0, 0.25, 0.5, 0.75, 1)$.
Tetapkan $a^1 = -50$ dan $b^1 = 50$.
Tetapkan $A^* = 0$, $X^* = 0$ dan F^*.

2). Tetapkan variabel chaos A_i^r ke variabel optimisasi $X_i^r \in [a^1, b^1]$ dengan

$$X_i^r = a^1 + A_i^r (b^1 - a^1) \quad (5)$$

3). Hitung fungsi objektif $F(X^r)$ dengan MSE
Jika $F(X^r) < F^*$ maka $A^* = A^r$, $X^* = X^r$ dan $F^* = F(X^r)$.
Jika $F(X^r) \geq F^*$ maka tinggalkan X^r

4). Hitung variabel chaos dengan

$$A_i^{r+1} = \mu\, A_i^r (1 - A_i^r) \quad (6)$$

5). Jika *r* belum sampai $r\ max$, maka ulangi 2 – 4. Jika sudah, maka lanjut ke 6.

Langkah 2) Petakan urutan chaotic ke interval variabel melalui gelombang pembawa sesuai dengan masalah yang diselesaikan.

6). Tetapkan *h max*.
Tetapkan $a^2 = -10$ dan $b^2 = 10$.
Tetapkan ρ = 0,4
Tetapkan variabel chaos A_i^h ke dalam variabel optimisasi $X_i^h \in [a^2, b^2]$ dengan

$$X_i^h = X_i^* + \omega \left(2 * A_i^h - 1\right), i = 1, 2, \ldots, n \quad (7)$$

Dan $\omega = \rho(b^2 - a^2)$, $\rho \in (0,\ 0.5)$

7). Hitung fungsi objektif $F(X^h)$ dengan MSE
Jika $F(X^h) < F^*$ maka $A^* = A^h$, $X^* = X^h$ dan $F^* = F(X^h)$.
Jika $F(X^h) \geq F^*$ maka tinggalkan X^h

8). Hitung variabel chaos dengan

$$A_i^{h+1} = \mu A_i^h \left(1 - A_i^h\right) \quad (8)$$

dan $h = h + 1$

9). Jika h belum sampai $h\ max$, maka ulangi 6 – 8 Jika sudah, maka COA diakhiri dan X^* adalah solusinya.

Langkah 3) Bandingkan nilai fungsi dari setiap urutan chaotic dan memilih nilai terbaik sebagai output ketika nilai fungsi tetap invarian dan tidak bertambah lagi.

2.3. Ridge Polynomial Neural Network (RPNN)

Langkah RPNN adalah sebagai berikut [13]
a). Tambah PSN
b). Nilai random untuk bobot dan bias
c). Berhenti jika belum sampai error atau maksimal iterasi maka lakukan d – i
d). Setiap pola data lakukan e – h.
e). Masukkan pola data
f). Hitung output PSN dengan

$$y = f\left(\prod_{i=1}^{N}\left[\sum_{j=1}^{n} w_{ij}x_j + w_i\right]\right) \quad (10)$$

g). Hitung delta bobot PSN dengan

$$\delta_l = \mu(d - y)y' \prod_{\substack{z=1 \\ z \neq l}}^{j} h_z \quad (11)$$

$$\Delta w_{kl} = \delta x_k$$
$$\Delta w_{0l} = \delta_l$$

h). update bobot dan bias

$$w_{kl}(baru) = w_{kl}(lama) + \Delta w_{kl} \quad (12)$$

i). Jika berhenti hitung output RPNN adalah jumlah output tiap PSN dengan

$$= f\left(\sum_{j=1}^{N}\prod_{i=1}^{j}\left(\sum_{k=1}^{n} w_{ijk}x_k + \theta_{ji}\right)\right) \quad (13)$$

3. Hasil dan Pembahasan

Penelitian menggunakan data pelatihan adalah 40 baris terdiri dari kunci public dan cyphertext. Data pengujian adalah 9 baris kunci public dan cyphertext. Data target pelatihan dan pengujian adalah kunci privat.

Pengujian 1 menggunakan Jaingan Pi-sigma order 3 dengan learning rate 0.17 , 0.11 dan 0.07 adalah

Tabel 1 Hasil Uji Dengan Learning rate 0.17

No	Hasil	Error
1	35088	0.02067832456585944
2	37815	0.2920597174941962
3	34700	0.05246726003722088
4	47134	0.4611128615201393
5	49877	0.2004231842477866
6	51501	0.47501143045659566
7	42113	0.25465026718970013
8	43827	0.12147748212176235
9	41555	0.11260031159483766

Tabel 2 Hasil Uji Dengan Learning rate 0.11

No	Hasil	Error
1	35156	0.018010377687265558
2	38126	0.27990709597998464
3	34866	0.05896720016639203
4	47137	0.46097449772068055
5	49159	0.22852209336245366
6	50688	0.44323033744535223
7	41629	0.27358941762464256
8	44368	0.10030580201865158
9	41176	0.09777972101052593

Tabel 3 Hasil Uji Dengan Learning rate 0.07

No	Hasil	Error
1	35338	0.010901051503307932
2	38882	0.2503140345538356
3	35041	0.06583691126758855
4	46947	0.46842411949583074
5	48242	0.26440309486238067
6	49848	0.4103493614688051
7	41239	0.28885918305372876
8	44215	0.10629708397024978
9	40703	0.07928778989797561

Berdasarkan pada Tabel 1, Tabel 2, dan Tabel 3, hasil pengujian yang dekat akurasi paling baik adalah baris No. 1 dan No. 3

Pengujian 2 menggunakan Jaingan Pi-sigma order 4 dengan learning rate 0.17 , 0.11 dan 0.07 adalah

Tabel 4 Hasil Uji Dengan Learning rate 0.17

No	Hasil	Error
1	34737	0.03441085300450855
2	36356	0.3491206892338071
3	34869	0.05909820420231406
4	43640	0.5977885205184934
5	49420	0.21830250280236418
6	48981	0.37645002418361106
7	39203	0.368499251629795
8	44423	0.09817965744964718
9	40841	0.08469502190042824

Tabel 5 Hasil Uji Dengan Learning rate 0.11

No	Hasil	Error
1	35071	0.021340090020814412
2	37505	0.3041710652251349
3	34865	0.0589466134482233
4	45228	0.5356497168056327
5	49896	0.19969170296404792
6	50474	0.43485735548889226
7	41262	0.2879484078669922
8	45099	0.0717135066517412
9	40970	0.08971403482350451

Tabel 6 Hasil Uji Dengan Learning rate 0.07

No	Hasil	Error
1	35240	0.014725872855610889

2	38322	0.27223267308186544
3	35058	0.06648810361464448
4	46445	0.48806280887724707
5	48619	0.24963096223870151
6	49979	0.4154646600284473
7	41132	0.29301458916512574
8	44518	0.09444461115764924
9	40745	0.08090294884715390

Berdasarkan pada Tabel 4, Tabel 5, dan Tabel 6, hasil pengujian yang dekat akurasi paling baik adalah baris No. 1 dan No. 3

Pengujian 3 menggunakan Jaingan Pi-sigma order 5 dengan learning rate 0.17 , 0.11 dan 0.07 adalah

Tabel 7 Hasil Uji Dengan Learning rate 0.17

No	Hasil	Error
1	35069	0.02141039498194991
2	37773	0.29368709923529535
3	34701	0.0525263141340012
4	47654	0.4407359337186
5	49836	0.20202404742487978
6	51356	0.46935436803684916
7	42274	0.24835438611407012
8	44175	0.10789012408631637
9	41463	0.10901684128645697

Tabel 8 Hasil Uji Dengan Learning rate 0.11

No	Hasil	Error
1	33359	0.08833243751642408
2	33359	0.4663954307174501
3	33359	7.1031525390545785E-6
4	33359	0.9999999958353435
5	33359	0.8466466526804742
6	33359	0.2347233610380799
7	33359	0.5971363578912386
8	33359	0.5310226115124397
9	33360	0.20800244190645026

Tabel 9 Hasil Uji Dengan Learning rate 0.07

No	Hasil	Error
1	35369	0.009688206437920344
2	38082	0.2816271084255453
3	35050	0.06617667218027289
4	45396	0.5290798882753989
5	48961	0.23625725893611393
6	49205	0.38519251780879293
7	40734	0.308599139681946
8	44044	0.11298785511189341
9	40666	0.07784934910710359

Berdasarkan pada Tabel 7, Tabel 8, dan Tabel 9, hasil pengujian yang dekat akurasi paling baik adalah baris No. 1 dan No. 3

4. Kesimpulan

kriptanalisis ElGamal menggunakan Algoritma Optimasi Chaos dan Ridge Polinomial Neural Network menghasilkan kunci privat dengan baik. Beberapa variable dapat mempengaruhi tingkat keberhasilan dan menghasilkan nilai error yang tidak bergantung pada perbedaan learning rate dan banyaknya order PSN. Hasil penelitian dari setiap table yang hampir mendekati nilai target kunci privat adalah setiap table dari baris No. 1 dan No. 3. Baris tersebut mendapatkan hasil kunci privat dengan nilai error paling kecil dari kunci privat yang lainnya. Sehingga hasil penelitian yang didapat berpengaruh terhadap nilai kunci privat yang paling kecil dimana mendekat iterasi yang tidak terjebak ke dalam minimum local. Hal ini menghasilkan iterasi yang konvergen.

Daftar Pustaka

[1] S. Annapoorna, S. K. Shravya, K. Krithika, A Review on Asymmetric Cryptography RSA and ElGamal Algorithm, International Journal of Innovative Research in Computer and Communication Engineering, Vol.2, Special Issue 5, October 2014, ISSN(Online): 2320-9801, ISSN(Print): 2320-97798, 2014.

[2] M. Fatma, H. Aya, S. S. Nahla, A. A. Fatimah, A Survey on Cryptography: comparative study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms, IEEE International Conference on Cyber Security and Cloud Computing (CSCloud), DOI 10.1109/CSCloud/EdgeCom.2019.00022. 2019.

[3] B. Qiu, Y. Yao, X. Zhang, Y. Chen, Iterative Composite Encryption Algorithm Based on Tea and Elgamal, World Congress on Computer Science and Information Engineering, 978-0-7695-3507-4/08 $25.00 © 2008 IEEE, DOI 10.1109/CSIE.2009.744, 2009.

[4] G. Rozaida, J. H. Abir, A. Dhiya, L. Paulo, Time series Prediction using Dynamic Ridge Polynomial Neural Networks, Second International Conference on Developments in eSystems Engineering, 978-0-7695-3912-6/09 $26.00 © 2009 IEEE DOI 10.1109/DeSE.2009.35, 2009.

[5] X. Chunbo, L. Lifen, Parallel Chaos Optimization Algorithm, International Conference on Electrical and Control Engineering, 978-0-7695-4031-3/10 $26.00 © IEEE DOI 10.1109/iCECE.2010.361, 2010.

[6] Y. Xiaohan, C. Shuyu, S. Chuan, and L. Yuwen, A Novel Class Imbalance-oriented Polynomial Neural Network Algorithm for Disease Diagnosis, IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Pages: 2360-2367, DOI Bookmark: 10.1109/BIBM52615.2021.9669281, 2021.

[7] L. Jie, and T. Junyong, Based Mutation Chaos Particles Swarm Optimization of Vanet Performance in Qos Constraint, 978-1-6654-9109-9/22/$31.00 ©2022 IEEE DOI 10.1109/ICCNEA57056.2022.00054, International Conference on Computer Network, Electronic and Automation (ICCNEA), 2022.

[8] Y. Zhanpeng, Z. Yong, W. Youlong, and S. Yuanming, Communication-Efficient Quantized SGD for Learning Polynomial Neural Network, Pages: 1-6, DOI Bookmark : 10.1109/IPCCC51483.2021. 9679419, IEEE International Performance, Computing, and Communications Conference (IPCCC), 2021.

[9] D. Abhijit, and C. E. V. Madhavan, *Public-key Cryptography : Theory and Practice*, Pearson, 2020.

[10] T. A. Ahmad, *Renewable Energy Systems*, Academic Press, 2021.

[11] W. Yishu, and W. Xiao, Analysis of Lyapunov in Game Dynamics, International Conference on

Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM), 978-1-6654-9858-6/22/$31.00 ©2022 IEEE DOI 10.1109/MLCCIM55934.2022.00086, 2022.

[12] Z. Li, and C. Junpeng, A Novel Chaos Sparrow Search Algorithm for TSP Problem, 17th International Conference on Computational Intelligence and Security (CIS), 978-1-6654-9489-2/21/$31.00 ©2021 IEEE DOI 10.1109/CIS54983.202 1.00088, 2021.

[13] E. I. Vugar, *Ridge Functions and Applications in Neural Network*, American Surveys and Monographs, 2021.

[14] M. Seyedali, *Handbook of Moth-Flame Optimization Algorithm: Variants, Hybrids, Improvements, and Applications*, CRC Press, 2022

.