Fidelity : Jurnal Teknik Elektro
Vol. 4, No. 2, Mei 2022, page 40-44

p-ISSN: 2686-3650
e-ISSN: 2686-3642

Kriptanalisis DES menggunakan Jaringan Syaraf Tiruan

Mundi Wiwit Kurniaga
R&D
PT. Cogindo DayaBersama
Sukabumi, Indonesia
mundiwk@cogindo.co.id

Azhari Yulianto
R&D
PT. Cogindo DayaBersama
Sukabumi, Indonesia
azhariwer@@cogindo.co.id

Tri Setya Aji Kumoro
R&D
PT. Cogindo DayaBersama
Sukabumi, Indonesia
antonius.aji48@@cogindo.co.id

***Abstrak*—Selama dua dekade sebelumnya, Kriptanalisis *block ciphers* telah mengalami banyak perubahan, dengan banyak teknik Kriptanalisis baru yang diperkenalkan. Di sisi lain, karena aplikasinya tersebar di beberapa bidang, Jaringan Syaraf Tiruan telah menunjukkan keandalan dan penerapan yang tinggi. Selain itu, metode efektif untuk mengimplementasikan perangkat keras dan perangkat lunaknya dibuat lebih cepat dan lebih mudah. Algoritma genetik untuk meningkatkan jaringan saraf telah terbukti menjadi cara yang menjanjikan untuk menghasilkan lebih baik dan lebih mudah untuk melatih jaringan saraf. Makalah ini memperkenalkan teknik Kriptanalisis baru berdasarkan jaringan saraf yang dihasilkan menggunakan algoritma genetika. Jenis kriptanalisis ini disebut sebagai serangan teks biasa yang diketahui. Metode yang diusulkan dimasukkan ke dalam perangkat lunak dan diuji terhadap *Data Encryption Standard* (DES). DES adalah *block cipher* yang canggih dengan ukuran blok 64-bit. Dalam hal waktu, kemampuan pemrosesan, dan jumlah pasangan *ciphertext-plaintext* yang diperlukan untuk Kriptanalisis, implementasi perangkat lunak sistem yang disarankan telah menunjukkan keuntungan yang signifikan dibandingkan sistem Kriptanalisis yang ada. Kemampuan pembelajaran jaringan syaraf tiruan dan aplikasinya dalam kriptanalisis menjadi pokok bahasan penelitian ini.**

Kata Kunci—Kriptanalisis, Jaringan Syaraf Tiruan.

I. PENGANTAR

Kriptografi adalah sarana penyimpanan dan pengiriman data dalam format yang hanya dapat dibaca dan diproses oleh mereka yang seharusnya. Ini adalah ilmu penyandian informasi ke dalam format yang tidak dapat dibaca untuk mengamankannya. Kriptografi adalah alat yang ampuh untuk melindungi data sensitif saat disimpan di media atau diangkut melalui rute komunikasi jaringan. Sebagai generalisasi model matematika sistem saraf biologis, jaringan saraf tiruan telah dibuat. Setelah McCulloch dan Pitts memperkenalkan neuron yang lebih dangkal pada tahun 1943, gelombang pertama minat pada jaringan saraf (juga dikenal sebagai model koneksionis atau pemrosesan terdistribusi paralel) muncul. Jaringan saraf tiruan (JST) adalah struktur pemetaan non-linier yang terinspirasi oleh fungsi otak manusia. Mereka adalah alat pemodelan praktis, terutama ketika hubungan data yang mendasarinya tidak jelas [18]. Jaringan saraf tiruan dapat mendeteksi dan mempelajari pola terkait antara set data input dan nilai target. Mereka dapat digunakan untuk memprediksi hasil data input independen baru setelah pelatihan. Properti nonlinier jaringan saraf diinginkan dalam kriptografi.

Dalam subjek kriptanalisis, kemampuan jaringan saraf untuk mencari ruang solusi dapat membantu. Mungkin juga dapat mengembangkan jenis serangan baru pada algoritme saat ini berdasarkan pada premis bahwa jaringan saraf dapat menduplikasi fungsi apa pun, sehingga memungkinkan untuk menentukan solusi yang tepat dan memecahkan metode. Kami mulai dengan motivasi untuk desain cipher blok dalam pekerjaan ini dan kemudian membangun algoritma DES di MATLAB. Mereka memiliki kriteria desain cipher blok yang paling sering digunakan. Kemampuan jaringan saraf untuk menjalankan operasi pemetaan yang rumit dari satu domain ke domain lain adalah kekuatan pendorong di balik konsep ini. Akibatnya, penelitian menggunakan kriptanalisis DES untuk menyelidiki properti nonlinier jaringan saraf.

II. LATAR BELAKANG

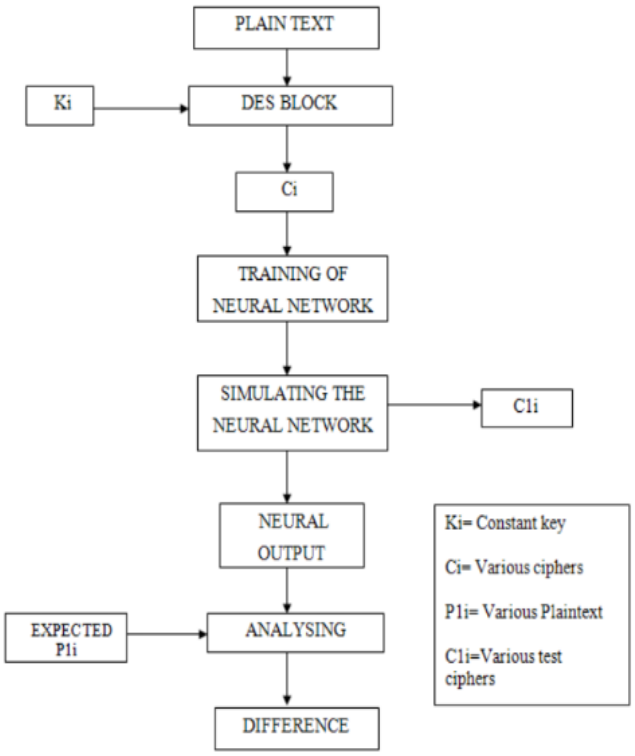
Beberapa teknik algoritma keamanan telah dikembangkan, secara kolektif dikenal sebagai *cipher*, yang digunakan untuk enkripsi dan dekripsi dan sulit untuk dipecahkan. Dalam beberapa tahun terakhir, *cipher* kontemporer telah menggunakan kunci untuk mengatur enkripsi dan dekripsi, dengan penerima hanya mendekripsi komunikasi jika kunci cocok dengan kunci enkripsi. Karena algoritma simetris menggunakan kunci yang sama untuk kedua operasi, penelitian tentang mekanisme kriptografi telah menunjukkan bahwa mereka lebih cepat dilakukan di komputer daripada algoritma asimetris [11]. Namun, kedua kunci (algoritma) digunakan untuk mengenkripsi dan mendekripsi data.

Ketika awalnya dimulai pada tahun 1977, *Data Encryption Standard* (DES), yang menggunakan blok dan kunci 64-bit, adalah standar enkripsi terkemuka untuk pengembangan aplikasi. Namun, karena metode ini rentan terhadap serangan *brute force*, lebih banyak *block cipher*, seperti *Advanced Encryption Standard* (AES), MARS, RC6, *Serpent*, dan *Twofish*, telah dibuat untuk meningkatkan keamanan data. Karena metode ini ramah pengguna terhadap kapasitas komputasi komputer, AES dipilih sebagai algoritma standar untuk keamanan internet dan perangkat keras. Namun, satu poin penting adalah bahwa algoritma ini menggunakan penjadwalan kunci, yang merupakan algoritma yang menghasilkan sub-kunci untuk setiap putaran yang diberikan kunci. Karena kriptanalisis diferensial dan linier mudah dilakukan ketika pembangkitan kunci lemah, penelitian telah dilakukan pada penerapan Jaringan Syaraf Tiruan (JST) konvensional untuk mengatasi masalah keamanan ini. Di sisi lain, *spiking Neural Networks*

(SNN) menggunakan neuron simulasi untuk setiap unit operasi dasar, dan waktu keluaran disimpan dalam 1s dan 0, dengan satu menunjukkan bahwa ada sesuatu yang terjadi dan 0 menunjukkan bahwa tidak ada yang terjadi. SNN menggunakan fungsi tanda sebagai fungsi aktivasi, kemudian memproyeksikan bobot dan koefisien filter untuk diubah, memberikannya lebih banyak independensi daripada JST.

Untuk mengatasi kesulitan kritis pribadi, itu diterapkan di SNN, blok sandi simetris prototipe dengan panjang blok 8-bit yang dapat diperbesar menjadi 64 bit, 128 bit, atau panjang variabel apa pun yang diperlukan. *Block cipher* yang digunakan untuk penelitian ini adalah *Simple-Data Encryption Standard* (S-DES), yang merupakan bentuk DES yang lebih pendek dan banyak digunakan untuk keamanan internet. Selain itu, teknik ini memungkinkan perubahan yang lebih cepat dalam kunci enkripsi dan enkripsi tingkat jaringan tanpa faktorisasi.

III. PENERAPAN

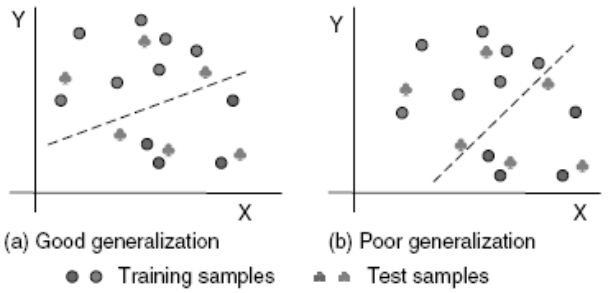


Gambar 1: Implementasi jaringan syaraf tiruan

Kami berpikir untuk memanfaatkan jaringan saraf untuk mengimplementasikan DES. Kami sedang berpikir tentang jaringan saraf *feedforward*. Di sini, jaringan saraf empat lapis digunakan untuk mengimplementasikan DES. Lapisan input jaringan saraf memiliki 64 neuron, tingkat menengah memiliki 100 neuron masing-masing, dan lapisan *output* memiliki 64 neuron.

Menggunakan blok DES dengan plaintext variabel dan kunci konstan, berbagai *cipher* dibuat, yang kemudian diumpankan ke blok saraf pelatihan dengan *plaintext* yang bervariasi. Gambar 1 menggambarkan *output* saraf yang diterima setelah mensimulasikan jaringan saraf ini dengan berbagai set cipher. Untuk kriptanalisis, keluaran saraf ini dibandingkan dengan teks biasa yang diprediksi.

Pelatihan dan pengujian jaringan saraf, Metode pelatihan yang paling efektif adalah dengan mengumpulkan sejumlah besar contoh (lebih banyak contoh diperlukan untuk masalah yang lebih rumit) yang menggambarkan semua sifat masalah yang berbeda. Dalam kasus lain, kebisingan atau keacakan lainnya dipasok ke data pelatihan untuk memperkenalkan jaringan dengan kebisingan dan fluktuasi data alami [4] untuk membangun jaringan yang tangguh dan dapat diandalkan. Data pelatihan yang tidak memadai pada akhirnya menghasilkan jaringan yang tidak dapat diandalkan dan tidak dapat diprediksi. Biasanya, jaringan dilatih untuk sejumlah *epoch* tertentu atau sampai kesalahan keluaran turun di bawah ambang batas kesalahan yang ditentukan. Sangat penting untuk menghindari tekanan berlebihan pada jaringan. Dengan *overtraining*, jaringan mungkin menjadi terlalu beradaptasi untuk mempelajari sampel dari set pelatihan, membuatnya tidak mampu mengklasifikasikan data secara efektif dari luar set pelatihan. Gambar 2 menggambarkan hasil kategorisasi jaringan yang telah di-*overtrain*. Tujuannya adalah untuk secara akurat mengkategorikan X dan Y. Pola ujian tidak ditampilkan selama pelatihan. Seperti diilustrasikan pada Gambar 2 (kiri), setiap kelas data uji telah diidentifikasi secara akurat, meskipun tidak terlihat selama pelatihan. Disebutkan bahwa jaringan yang dilatih memiliki kinerja generalisasi yang kuat. Beberapa kesalahan representasi dari hasil pengujian terlihat pada Gambar 2 (sebelah kanan). Jaringan pertama-tama belajar untuk mengidentifikasi properti global dari input dan menggeneralisasi dengan sangat baik sebagai hasilnya. Namun, setelah pelatihan yang cukup, jaringan mulai membedakan pasangan input/output tertentu, sebagai lawan dari penentuan bobot yang mencirikan pemetaan untuk keseluruhan set pelatihan dalam pengertian umum.



Gambar 2: Ilustrasi kinerja generalisasi

menentukan jumlah neuron, Jumlah neuron tersembunyi mempengaruhi kemampuan jaringan untuk memisahkan data. Sejumlah besar neuron tersembunyi memastikan pembelajaran yang tepat, dan jaringan dapat secara akurat memprediksi data yang dilatihnya, tetapi kinerjanya pada data baru dan kapasitasnya untuk menggeneralisasi menurun. Dengan jumlah neuron tersembunyi yang tidak mencukupi, jaringan mungkin tidak dapat menemukan korelasi antara data, dan kesalahan tidak akan berkurang ke tingkat yang dapat diterima. Oleh karena itu, memilih jumlah neuron yang terkubur adalah pilihan yang signifikan.

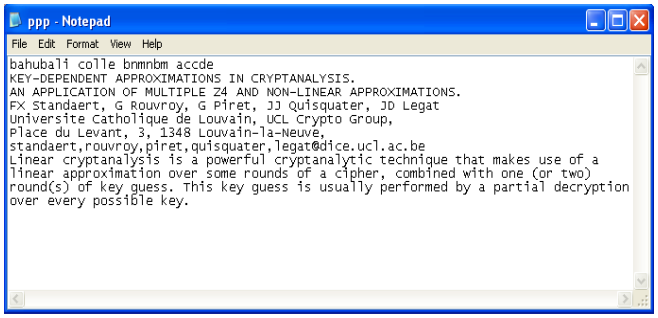
Memilih bobot awal, Algoritma pembelajaran menggunakan metode penurunan paling curam, yang menggulung ruang berat lurus ke bawah hingga lembah pertama tercapai. Titik awal pertama dalam ruang berat

multidimensi sangat penting. Namun, tidak ada kriteria yang disarankan untuk keputusan ini selain bereksperimen dengan berbagai nilai bobot awal untuk melihat apakah kinerja jaringan dapat ditingkatkan.

Memilih kecepatan belajar, Ketika setiap bobot diperbarui, kecepatan pembelajaran secara efektif menentukan jumlah langkah yang dibuat dalam ruang bobot multidimensi. Jika tingkat pembelajaran yang ditetapkan terlalu tinggi, minimum lokal dapat terus dilampaui, menyebabkan osilasi dan konvergensi lambat ke keadaan kesalahan yang lebih rendah. Jika kecepatan pembelajaran tidak mencukupi, jumlah iterasi yang diperlukan mungkin berlebihan, sehingga menghasilkan kinerja yang buruk.

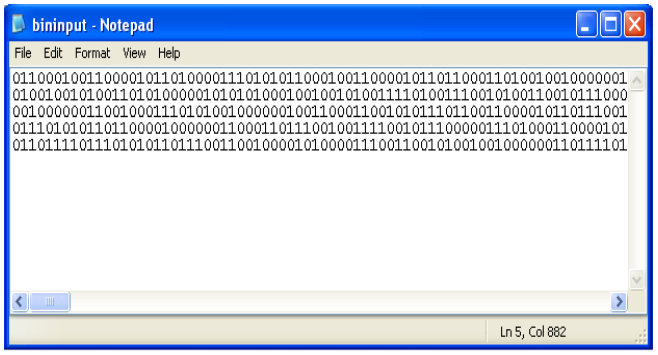
IV. HASIL

Teknik *Data Encryption Standard* digunakan untuk membangun *ciphertext* dari *plaintext* dan kunci konstan dalam pekerjaan ini. Di sini, seorang programmer harus membaca file yang berisi string karakter dan kemudian membuat file biner dengan konten yang sama. Di sini, file 'ppp.txt' dibaca, yang berisi serangkaian karakter seperti yang ditunjukkan pada Gambar 3, dan 'bininput.txt' dihasilkan, yang merupakan versi biner dari file 'ppp.txt' seperti yang ditunjukkan pada Gambar 4.



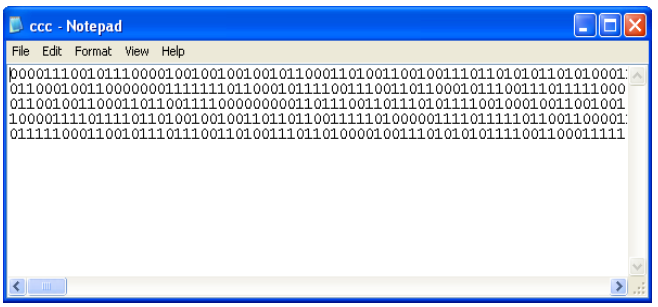
Gambar 3: File yang terdiri dari sekumpulan karakter

Saat mengonversi file teks ke mitra binernya, masing-masing dari delapan karakternya dapat diekstraksi satu per satu. Satu karakter dapat menghasilkan delapan bit biner, maka kombinasi dari delapan karakter menghasilkan 64 bit. Selama proses enkripsi dan dekripsi, 64 bit dan kunci konstan 64 bit ini digunakan secara bersamaan. Untuk enkripsi dan dekripsi dalam proyek ini, kunci kontinu E326D1475398000 (dalam heksadesimal) digunakan.



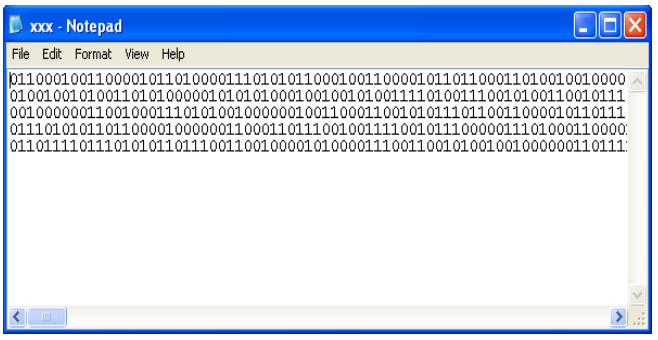
Gambar 4: File setara biner

Setelah membuat file ekuivalen biner 'bininput.txt', file tersebut dapat dibaca kembali dan file lain 'ccc.txt' berisi bit biner terenkripsi yang dikenal sebagai bit ciphertext dapat diproduksi menggunakan fungsi enkripsi, bit plainteks, dan kunci, sebagai diilustrasikan pada gambar 5. File ini memiliki jumlah bit yang sama dengan file 'bininput.txt'.



Gambar 5: File *ciphertext*

Kemudian, file bernama 'xxx.txt' dapat dibuat dengan membaca 'ccc.txt' sebagai file dekripsi yang terdiri dari bit biner yang telah dienkripsi. Ini menggunakan fungsi dekripsi, bit ciphertext, dan kunci. File ini berisi bit yang sama dengan file 'bininput.txt' yang terlihat pada Gambar 6.



Gambar 6. File yang didekripsi

Algoritma jaringan saraf dilatih menggunakan file-file ini sebagai input dengan mengekstraksi bit dari file plaintext dan *ciphertext*. Saya menggunakan sekitar setengah dari bit yang dibuat dalam file-file ini. Di antara bit ciphertext dan bit plaintext yang dipilih, sampel simulasi diperoleh. Untuk berbagai menjalankan perangkat lunak pelatihan jaringan saraf ini, bagian yang berbeda digunakan. Sebagai contoh untuk setiap *run*, *ciphertext*, dan *plaintext* dipilih yang sesuai. Sebagai nilai input dan output, variabel p1 dan t1 masing-masing mewakili *ciphertext* dan *plaintext*. Beberapa contoh ditunjukkan di bawah ini.

Contoh 1((Di sini matriks p1 dan t1 berdimensi 8×64)

p1=[1 0 1 0 1 1 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 1
0 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 1 1 0
1 0 ;

0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 0 1 1 0 0 1 1 0 0 0 1 0 1 0 0 0 0
1 1 1 1 0 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 0 1;

0 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 1 0 0 1 0 1 0 1 1 1
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 1 0 0 1 1 1 0 0 1 ;

0 0 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 1
1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 ;

1 0 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 0 0
0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 1 1 1 0 0 1 ;

0 1 0 1 1 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0
0 1 0 0 0 1 1 0 0 1 1 0 0 1 0 1 1 1 1 0 1 1 1 0 0 0 0 1 1 1 0 1;

0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0
0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 ;

0 1 0 0 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 1 1 1
0 1 1 1 1 0 0 0 1 0 1 0 1 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 1 1 1
0];

t1=[0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1
0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 1 0 1 1
1 0 ;

0 1 1 0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 0
0 1 1 1 0 0 1 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 ;

0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0
0 1 1 0 1 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 0 0 1 1 0 1 0 0 1 ;

0 1 1 1 0 0 0 1 0 1 1 1 0 1 0 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 ;

0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0 1 1 0 0 1 1 0 0 0 0 1
0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 ;

0 1 0 1 0 1 0 1 0 1 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 1 0 0 0 0 0
0 1 0 0 0 0 1 1 0 1 1 1 0 0 1 0 0 1 1 1 1 0 0 1 0 1 1 1 0 0 0 0 ;

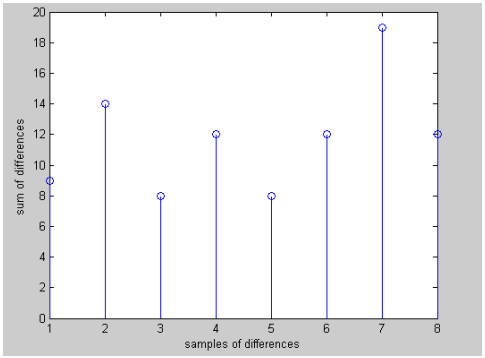
0 1 1 1 0 1 0 0 0 1 1 0 1 1 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 1
0 1 1 1 0 0 1 0 0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0 0 0 0 ;

0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0
0 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 0 1]
;

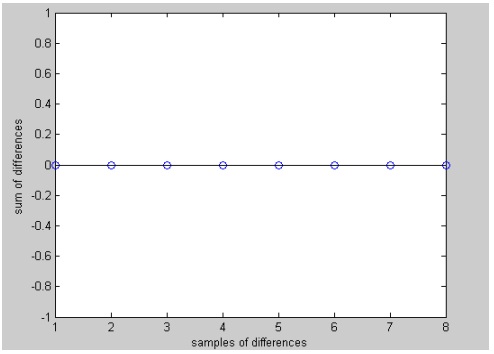
Setelah simulasi, keluaran jaringan dihasilkan bersama dengan grafik berikut yang ditunjukkan pada Gambar 7, di sini nilai rata-rata yang dihitung ditemukan sebagai 11.7500. Akhirnya, kesalahan kinerja dapat dihitung dengan menggunakan persamaan yang diberikan di bawah ini:

$$\textit{Performance Error} = (\textit{mean}/64) \times 100 \quad (1)$$

Oleh karena itu, kesalahan kinerja yang terbukti adalah 18,3594. Dengan urutan tersebut, fungsi aktivasi tersebut adalah tansig, logsig, purelin, dan satlin. Ini adalah salah satu kombinasi fungsi aktivasi program pelatihan jaringan saraf dengan kesalahan kinerja yang paling signifikan.



Gambar 7: Mewakili Jumlah Perbedaan



Gambar 8: Mewakili jumlah perbedaan

Perhatikan Gambar 8, di mana nilai rata-rata yang dihitung adalah 0. Oleh karena itu, kesalahan kinerja yang ditemukan di sini adalah 0. Dalam urutan itu, fungsi aktivasi ini adalah logsig, tansig, satlin, dan purelin. Ini adalah salah satu kombinasi fungsi aktivasi program pelatihan jaringan saraf dengan kesalahan kinerja terendah.

V. KESIMPULAN

Saat kita maju menuju peradaban di mana sumber daya informasi otomatis lebih banyak dibagikan, signifikansi kriptografi akan terus meningkat. Jaringan elektronik untuk perbankan, ritel, manajemen inventaris, pemberian manfaat dan layanan, penyimpanan dan pengambilan informasi, pemrosesan terdistribusi, dan aplikasi pemerintah akan memerlukan kontrol akses dan perlindungan data yang ditingkatkan. Algoritma DES telah menjadi usaha yang sukses dalam pengembangan awal sistem keamanan. Ini akan terus menjadi algoritma kripto yang paling banyak diperiksa, diuji, dan digunakan untuk beberapa waktu. Tapi mungkin kontribusi paling signifikan dari DES adalah bahwa hal itu telah menyebabkan pertimbangan keamanan tambahan yang harus diambil untuk memiliki sistem dan jaringan komputer yang aman.

Saya mengusulkan penggunaan jaringan syaraf tiruan untuk kriptanalisis dalam penelitian ini. Saya menyelesaikan kriptanalisis DES praktis menggunakan plaintext dan kuncinya. Di sini, kesalahan kinerja yang ditetapkan digunakan untuk mengevaluasi data. Akibatnya, efektivitas dapat diperoleh. Dengan meningkatkan jumlah sampel yang digunakan untuk melatih jaringan saraf dan mengubah bobot dan bias neuron di setiap lapisan, efisiensi dapat ditingkatkan lebih lanjut.

Banyak pekerjaan di masa depan pada pemilihan bobot jaringan saraf dan adaptasi (pelatihan) tetap ada. Opsi implementasi, terutama dalam hal perangkat keras, masih belum dijelajahi. Kriptanalisis dari enkripsi yang disarankan dan peningkatan keamanannya adalah bidang penelitian yang berbeda di masa depan. Berbagai jenis jaringan saraf dapat digunakan untuk kriptanalisis.

REFERENSI

[1] William Stallings, “Cryptography and Network Security”, 3 edition, Pearson Education (Asia) Pvt.Ltd./Prentice Hall of India, 2003.

[2] Dr. K. S. Ooi and Brain Chin Vito, “Cryptanalysis of S-DES”, 2002.

[3] Carlos Gershenson, “Artificial Neural Networks for Beginners”

[4] Ajith Abraham, “Artificial Neural Networks ” [5] Fiona Nielsen, “Neural Networks – algorithms and applications”

[5] Rudra Pratap, “Getting started with Mat Lab 6.0”

[6] K.V .Srinivas Rao , M. Rama Krishna and D. Bujji Babu, “Cryptanalysis of a feistel Type Block Cipher by Feed Forward Neural Network Using Right Sigmoidal Signals”,2009

[7] Howard M. Heys ,“A Tutorial on Linear and Differential Cryptanalysis” Memorial University of Newfoundland, email: howard@engr.mun.ca

[8] Anders Krogh, “What are artificial neural networks? ”

[9] Simon Haykin , “Feedforward Neural Networks : An Introduction”

[10] Craig Smith ,“Basic Cryptanalysis Techniques” , November 17th, 2001.

[11] Khaled M. Alallayah , Waiel F. Abd El-Wahed, Mohamed Amin and Alaa H. Alhamami , “Attack of Against Simplified Data Encryption Standard Cipher System Using Neural Networks”, Journal of Computer Science 6 (1): 29-35, 2010

[12] RC Chakraborty , “Fundamentals of Neural Networks”, June 01 ,2010

[13] Subariah Ibrahim,Mohd Aizaini Maarof, “A Review On Biological Inspired Computation In Cryptology”, Universiti Teknologi Malaysia, Skudai 81300, Johore, Malaysia.

[14] www.infosecuritymag.com/articles/july01/columns_logoff.shtml

[15] Kaisa Nyberg , “Methods for Symmetric Key Cryptography and Cryptanalysis”, 2009.

[16] S N Sivanandam, S Sumathi, S N Deepa, “Introduction To Neural Networks Using MATLAB 6.0”,2006

[17] Girish Kumar Jha, “Artificial Neural Networks And Its Applications”, I.A.R.I, New Delhi, girish_iasri@rediffmail.com

[18] Reaffirmed,“Data Encryption Standard (DES)”, U.S. Department Of Commerce/National Institute of Standards and Technology, 1999 October 25

[19] Biham, E. and Shamir, A.:“A Differential Cryptanalysis of the Data Encryption Standard”, Springer Berlin Heidelberg New York, 1993.

[20] H. Feistel, "Cryptography and computer privacy," Sci. Amer., vol. 228, no. 5,pp. 15-23, May 1973.

[21] W. Diffie, M. Hellman, "Exhaustive cryptanalysis of the NBS data encryption standard," Computer, pp. 74-78, June 1977.

[22] Martin Albrecht and Carlos Cid, "Algebraic Techniques in Differential Cryptanalysis” Information Security Group, Royal Holloway, University of London Egham, Surrey TW20 0EX, United Kingdom