Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 6, 2025

PERFORMANCE IMPROVEMENT OF SAC USING NEW EMD AND MD BASED 2D-OCDMA

Hamza Mumtaz[1], Khalid Rehman[*2], Kiran Raheel[3], Adil Nawaz[4], Hashim Ali[5]

[1,*2,3,4]*Department of Electrical Engineering, CECOS University of Information Technology & Emerging Sciences, Hayatabad, Peshawar.*
[5]*Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan.*

[*2]khalid@cecos.edu.pk

DOI: https://doi.org/10.5281/zenodo.15680729

Keywords
Spectral Amplitude, Coding-Optical, Code Division Multiple Access, Multiple Access Interface, Multi-diagonal, Enhanced Multi-diagonal Codes, Bit Error Rate.

Article History
Received on 10 May 2025
Accepted on 10 June 2025
Published on 17 June 2025

Corresponding Author: *
Khalid Rehman

Abstract
The adaptability of a passive optical network necessitates the employment of an appropriate multiple-access method that can offer the requisite transmission capacity in terms of data, reach, and number of users while being simple to construct and low in cost. Because of its asynchronous nature and simultaneous access to the channel for different users, the spectral amplitude coding- photonic code division multiple access system is expected to offer the needed capacity. However, in high cardinality systems, the 1D character of the spectral amplitude coding method restricts the reduction of multiple access interference and the related phase-generated intensity noise. In addition, dividing available spectral windows restricts support for large cardinality in zero or fixed-phase cross-correlation systems. As a result, for high transmission capacity and a large number of users across a long distance, a new dimension must be added to the existing 1D code. To include spatial encoding, currently employed 2D spectral amplitude coding-optical code division multiple access systems use spectral/spatial coding techniques that necessitate a significant number of optical fiber media between the transmission and reception modules. This severely affects the practicality of implementing 2D optical code division multiple access in a passive optical network. As a result, spectral/temporal coding has been optimized for low-cost passive optical networks. To support a high number of users while maintaining a low bit error rate, spectral/temporal coding is necessary. Based on current 1D multidiagonal and improved multidiagonal codes, this work proposes a unique 2-D spectral/temporal coding method. We do a thorough mathematical study using bit error rate, quality factor, and eye diagram as performance metrics. The system's performance shows that the code is efficient in terms of user count, multiple access interference, and encoder/decoder architecture.

INTRODUCTION

When compared to conventional time division multiple access (TMDA) and frequency division multiple access (FMDA), optical code division multiple access (OCMDA) is a promising technique because it provides high security, dynamic bandwidth assignment, asynchronous channel access, and support for multimedia applications [1]. By combining orthogonal code sequences, OCDMA allows users to access the same media at the same time. Non-coherent OCMDA has a simple and low-cost architecture, but coherent OCDMA has a greater performance. SAC is a good approach for finding information in non-coherent OCMDA by employing orthogonal code sequences [2]. SAC has lately attracted a lot of interest due to its ease of installation and ability to attenuate noise. SAC blocks enable the use of spectrum components of broadband light sources known as spectral chips, which are typically based on the orthogonal code sequence supplied to each user. As a result, code design is critical in noise suppression. Several codes are proposed for the SAC-OCDMA system, however these codes have disadvantages such as long code sequences, parameter selection, non-ideal cross-correlation, and user count [3]. The number of users increases multiple access interference (MAI), receiver complexity, and system throughput. SAC-OCDMA performance is restricted by short noise, thermal noise, and, most importantly, MAI and related phase intensity-induced noise (PIIN).

A coding system with low cross correlation can assist minimise overlapping spectra and, as a result, the PIIN. By combining additional temporal or geographical dimension with the current spectral dimension, 2-D codes for SAC- OCDMA can lower MAI. Multiple optical fibres and star couplers are employed to extend into spatial dimension for 2-D spectral/spatial coding, where the system complexity grows linearly with code weight and code length [1].

To add spatial encoding, currently utilised 2-D SAC-OCMDA systems use spectral/spatial coding techniques that need a significant number of optical fibre media between the transmission and reception modules. This severely affects the practicality of implementing 2-D OCDMA on a passive optical network (PON). As a result, spectral/temporal coding has been modified for low-cost PON. To handle a high number of users while maintaining a low bit error rate (BER), spectral/temporal coding is necessary.

Based on the phase information exchanged with the receiver, OCDMA detection is classified as coherent or non-coherent. SAC-OCDMA is a non-coherent spread spectrum technology that uses a coding system to transmit/block spectral chips [4].

Coherent and Incoherent OCDMA

In inconsistent OCDMA, a transmitter encodes using an ultrashort spectrum light source, and the same phase information is exchanged with the receiver to decode. The bipolarity of a coherent system allows for tight orthogonality, which eliminates MAI across different users. Coherent OCDMA is classified into two types: Spectral Phase Coded (SPC)-OCDMA and Temporal Phase Coded (TPC)-OCDMA. Because the phase of a coherent system must be synchronized, the system complexity increases [5]. In general, synchronous schemes with detailed transmission plans yield greater throughput than asynchronous schemes when system access is random and collision occurs.

In incoherent OCDMA with direct or balanced detection, intensity modulation is utilized. Incoherent systems are unipolar because they only send "Ones" and "Zeros." The performance of an incoherent system is lower than that of a coherent system, but it offers simplicity, cost-effectiveness, no phase constraint, and other benefits. As a result, incoherent systems are becoming more prevalent in PON and Sensor Networks (SN).

Incoherent access is classified into three types: temporal spreading, spatial coding, and SAC) [2]. A single bit is broken into tiny duration chips equal to the code length in temporal spreading, and the chips are repeated for each bit code-weight times to match the code family. Due to the high number of users and the weight of various users, chip overlapping occurs, resulting in MAI at the receiving end. In contrast, numerous cores of single-mode fiber are employed in spatial coding. Researchers proposed 2D codes such as temporal/spatial codes and spectral/spatial codes. Spatial codes have the drawback of having an equal light path and requiring a large amount of optical fiber, which makes this technology expensive [6]. In contrast, with SAC, wideband light is transmitted and filtered according to the coding system. To extract the information, it employs an appropriate detection approach. The SAC system presents a simple, low-cost, incoherent optical source transmitter and receiver configuration. By minimizing MAI and improving system performance, SAC- OCDMA is the most suited system for optical multi-access method over other OCDMA systems. The design of better codes with essential auto- and cross-correlation features is one of the profitable techniques for reducing interference.

Literature Review

In wireless communication, Code Division Multiple Access (CMDA) provides various advantages. It has inspired researchers to implement it in the optical domain. Viterbi demonstrated the viability of CDMA in the optical domain in 1986. In 1989, Salehi evaluated and analyzed the technique's performance and proposed the Optical Orthogonal Code (OOC) for Optical CDMA [7].

Yadav et al. presented the prime sequence codes with limited hamming space of P-1, where P is a prime integer in [8]. Prime code may be built in two steps: first, for a prime integer P and the Galois Field GF(P), create prime sequences. Second, previously created prime sequences are mapped onto binary sequences. The length of codeset P2 and the weight of code P both contain P prime sequences. The codeset of the preceding generating method for prime number P can provide the P number of codes. Furthermore, the number of concurrent active users is determined by the correlation features of these codes while maintaining a minimal BER. The number of concurrent active users can be raised by raising the prime number P. The code sequences' autocorrelation is so low that it is less than the number P, and the highest cross-correlation is two. The author also described the extended prime code, which increases the length of the code sequences considerably. The updated code has lowered the cross-correlation to one, which improves the MAI if the sequence autocorrelation is the same as the original.

Chung et al. presented optical orthogonal codes with family 071 that had good autocorrelation and cross-correlation features in [9]. The peak of autocorrelation is strong, but the peak of cross-correlation is low between any two sequences. Because the orthogonal code is incomplete, the code is known as pseudo orthogonal code. The cyclic shift in the code may not affect the sequences' correlation characteristics. Cyclic shift increases the cardinality of the code, allowing for a greater number of users.

Wong et al. presented a novel prime code called modified prime codes for synchronous mode in [10]. Many codes are recommended for asynchronous mode; however, because the code is also useable in synchronous mode, it outperforms the rest of the prime codes. Because the codes may be utilized in synchronous transmission, they have higher code cardinality and performance than asynchronous codes. We may witness a significant rise in the number of prospective subscribers and simultaneous users in the OCDMA network as a result of the cyclic shift in code generation. By cyclically changing the initial prime code sequence P times, MPCs can be generated. The number of code sequences can therefore be expanded by the original prime code, resulting in enhanced cardinality.

In [11], the author suggested the MQC/MQC code for the wavelength/time scheme, as well as the transmitter design and balanced structure for the receiver design, using Tunable Optical Fibre Delay Line (TOFDL) for the time dimension and Fibre Brag Granting (FBG) for the wavelength dimension. The design calculated BER by counting PIIN, shot noise, and thermal noise and then verifying the results with simulation. The system supports more users, requires a low-power signal for the light source, and uses bandwidth more effectively, although the MQC code development is difficult.

Based on test findings using matrix codes, the author in [12] presented a system of 2D temporal/spatial (T/S) with non-coherent OCDMA. The framework is implemented by separating multimode optical couplers and delay lines. This code is being investigated provisionally based on pseudo-symmetrical matrix codes with one pulse per row, weight equivalent to four, and four users. The author demonstrated that loss of the suggested plan is less than temporal system misfortunes because the number of couplers is less; there is no side lobe in autocorrelation and the consistency of the coupler part is critical for a legitimate relationship; and a shorter piece time can be used for a given arrangement of laser beam width.

In [13] proposes a class of codes known as 2D projection codes with balanced detection, and their performance is compared to that of 3D codes. This code's codeset is small in comparison to the codeset of 3D codes. It primarily uses balanced detection to eliminate the various access impedances. This code is created by predicting 2D sub-code words onto the 1D wavelength hopping code. The number of users for a particular codeset can be raised by relaxing correlation restrictions with a significant reduction in BER.

The authors of [14, 15] introduced the wavelength hopping/time spreading approach with the prime code and studied the security against eavesdroppers in the optical network. The approach employs a WDM/TDM hybrid to boost network capacity. When there is no time shift, the autocorrelation is zero, and when there is a phase shift, the autocorrelation is maximum. The code's greatest potential cross-correlation is unity. MAI is lower when numerous users speak at the same time because cross-correlation is lower when compared to other codes.

The authors in [16] propose a 2D wavelength/time code that uses the optimal Golomb ruler approach on the wavelength and time dimensions. The number of time slots is reduced, as is the number of concurrent users, although utilizing an optical hard limitation and guard time can enhance this. The codeset has grown in cardinality and ISD. For local area networks (LAN), the code employs a unipolar style with intensity and direct modulators as transceivers.

In [17], the author employed 3D coding with a single pulse per plan for SAC-OCDMA with a deferential detecting system. The code construction employs a 1D Golomb Ruler sequence, which has a higher cardinality and demonstrates improved BER performance. The cardinality of the codeset of a single pulse per plane is greater. For the elimination of the MAI, the deferential detection system is applied. The code has a cross-correlation of one and an out-of-phase autocorrelation of zero. The code employs bipolar assignment for each user, which implies that each user sends one for "high bit" and another for "low bit," resulting in a user count equal to half of the codeset sequences.

Contribution

Because of its asynchronous nature and simultaneous access to the channel for numerous users, the SAC-OCDMA system is expected to deliver the needed capacity. However, the 1D character of the SAC scheme restricts the lowering of MAI and the accompanying PIIN in systems with high cardinality. Furthermore, dividing available spectral windows restricts support for large cardinality in zero or fixed in phase cross-correlation systems. As a result, in order to support high transmission capacity and a large number of users across a long distance, a new dimension must be added to the existing 1D code.

To add spatial encoding, currently utilized 2D SAC-OCMDA systems use spectral/spatial coding techniques that need a significant number of optical fiber media between the transmission and reception modules. This severely lowers the practicality of implementing 2D OCDMA at the PON. As a result, spectral/temporal coding has been modified for low-cost PON. To support a high number of users while keeping a low BER, a code for spectral/temporal scheme is required.

Based on current 1D Multidiagonal (MD) and Enhanced Multi-Diagonal (EMD) codes, this work proposes a unique 2-D spectral/temporal coding method. We do a thorough mathematical study using BER, quality factor (Q-Factor), and eye diagram as performance metrics.

Paper Organization

The rest of the paper is organized as follows: First, we look at the fundamentals of code construction, delving into the programming languages, tools, and processes used. Following that, we go into the complexities of developing the 2D Empirical Mode Decomposition/Molecular Dynamics (EMD/MD) code, putting insight into the algorithms and computational approaches employed. We provide and analyze the results of experiments and simulations performed using the code in the Results & Discussions area, providing useful insights and conclusions. Finally, the article finishes with a References section, which catalogs all mentioned sources and background literature for reference and future research, as is common in academic writing.

Code Construcion

The EMD/MD code may be derived from existing EMD and MD codes, with each code denoted by (N, ω, λc). Where N represents code length, ω represents code weight, and λc represents phase cross-correlation of any two distinct sequences.

Multi Diagonal (MD) Code Construction

MD code is created such that any code weight larger than two can be chosen at random. Where the code length is determined by the number of users and the code weight. Equation 1[18] is used to execute the MD code creation.

$$S_{i,j} = \begin{Bmatrix} (K_y + 1 - i), & \text{for } j = \text{Even number} \\ i, & \text{for } j = \text{Odd numbers} \end{Bmatrix} \quad (1)$$

The series of diagonal matrices S(i,j) is constructed as follows for the number of users Ky=4 and code weight Wy=2. Where i=1,2,...Ky and j=1,2,...Wy.

$$S_{i,1} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, S_{i,2} = \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

The diagonal matrices S(i,j) are then converted to get T(i,j), where each element of S(i,j) represents a "One" position.

$$T_{i,1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{i,2} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

All binary matrix combinations T(i,j) constitute the MD code for K_y×ω_y is the final code for K users and W code weight. Equation 2 [18] may be used to calculate code length N.

$$N = K_y \times \omega_y \quad (2)$$

$$MD = [T_{i,1} | T_{i,2}] \quad (3)$$

$$MD = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}_{K \times N}$$

MD coding has no cross-correlation and is capable of mitigating MAI. The code provides flexibility in terms of code weight, number of users, and system design simplicity. It has a greater user base than Modified Quadratic Congruence (MQC) and Random Code (RD).

EMD Code Construction

EMD code is divided into two segments: data and code. D_(K_x)and C_(K_x) signify the data matrix and code matrix, respectively. The EMD code is represented by (N,ω,λ_c), the data matrix, is a square matrix of K_x× K_x. The cross-correlation between any two rows of the D_(K_x) is zero, the matrix's code weight is one, and entries on the major diagonal are one [19].

$$c = [d_{i,j} = 0 \text{ for } i \neq j] \quad (4)$$

Where i,j∈{1,2,…,K_x}

For example, the data matrix is designed as per equation 4 for the code weight of one and four users as;

$$D_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The size of the code matrix C_(K_x) is K_x× J, where K_x indicates the active number of users and J is the length of the code. The code matrix's attributes are as follows. The cross-correlation between any two adjacent rows is always one, and the code matrix C_(K_x) has a weight of two. The code matrix is provided for the four users as;

$$C_4 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Equation (5) may be used to get the code matrix code length.

$$J = K_x(\omega - 2) + 1 \quad (5)$$

Combining both sections, the Data matrix and the Code Matrix yields the EMD code [19].

$$EMD = [D_{K_x} | C_{K_x}]_{K_x \times N} \quad (6)$$

$$EMD(9,3,1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Development Of 2d Emd/Md Code

If we assume that the EMD code set and MD code set are represented with X and Y respectively. Then X_g and Y_h are any code sequences from the EMD and MD code set and has the elements $X_g=[x_g(1),x_g(2),\dots,x_g(N_x)]$ and $Y_h=[y_h(1),y_h(2),\dots,y_h(N_y)]$. The N_x

and N_y are the code lengths of code sequence X_g and Y_h respectively [17]. In EMD/MD code, X_g, represents the spectral components and MD code, Y_h, represents the temporal components.

$$A_{g,h} = Y_h^T X_g \tag{7}$$

$A_{(g,h)}$ represents the code word of each user such that the wavelength-hopping pattern along the X_g code and time-spreading pattern along the Y_h code. The newly developed code can accommodate N number of users and can be found as [17].

$$N = N_X \times N_Y \tag{8}$$

Table 1: EMD Code Set,X $, for\ \omega_x = 2\ \&\ K_x = 4$

X_1	1	0	0	0	0	0	0	1	1
X_2	0	1	0	0	0	0	1	1	0
X_3	0	0	1	0	0	1	1	0	0
X_4	0	0	0	1	1	1	0	0	0

Table 2: MD Codeset, $Y, for\ \omega_y = 2\ \&\ K_y = 4$

Y_1	1	0	0	0	0	0	0	1
Y_2	0	1	0	0	0	0	1	0
Y_3	0	0	1	0	0	1	0	0
Y_4	0	0	0	1	1	0	0	0

		X_1									X_2									X_3									X_4								
		1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0
	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0
Y_4	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

For the four number of users K=4, in 1D EMD code, X and the MD code, Y, the above code set can be generated. Any two arbitrary code sequences X_g and Y_h can be selected each from one code set to obtain the 2D EMD/MD code. An example of the 2D EMD/MD code set $A_{(g,h)}$ is shown below where code sequence X_g from EMD and Y_h from the MD code is used.

The cardinality of X_g code is C_x and the cardinality of Y_h code is C_y, so the cardinality of the new 2D code will be [20]

$$C = C_X \times C_Y \tag{9}$$

Cross Correlation Properties of the 2D code

To obtain the cross correlation of the newly developed 2D code, the following characteristics matrices are defined [21].

$$A^1_{g,h} = Y_h^T X_g \tag{10}$$

$$A^2_{g,h} = Y_h^T\, \overline{X_g} \tag{11}$$

The characteristics matrices are denoted with $A^d_{(g,h)}$ such that $d\in\{1,2\}$ for code word $A_{(g,h)}$ and $(X_g)^{-}$ is the complementary sequences of the X_g. The cross correlation of any two random code sequences is found using the following equation [22].

$$R^d_{m,n}(g,h) = \sum_{i=1}^{N_X}\sum_{j=1}^{N_Y} a^d_{m,n}(i,j)a_{g,h}(i,j) \tag{12}$$

Where $a^d_{(m,n)}$ shows $N_x \times N_y$ matrix of the characteristic matrices $A^d_{(m,n)}$ and $a_{(g,h)}$ is the N_x xN_y matrix belongs to $A_{(g,h)}$. The cross correlation of any two random code word $A_{(g,h)}$ and $A_{(m,n)}$ can be calculated as presented in the table below.

Table 3: Cross Correlation of 2D EMD/MD code

	$R^1_{m,n}(g,h)$	$R^2_{m,n}(g,h)$
$g = m = 1, h = n = 1$	$\omega_X\omega_Y$	0
$g = m = 1, h = n \neq 1$	ω_Y	$(\omega_X - 1)\omega_Y$

Where ω_X and ω_Y are the code weight of the code X and Y respectively. Using the following cross correlation property, the MAI can be cancelled [11].

$$R^1_{m,n}(g,h) - \frac{1}{\omega_Y - 1}R^2_{m,n}(g,h) = \begin{cases}\omega_X\omega_Y & \text{for } g = m = 1, h = n = 1 \\ 0 & \text{Otherwise}\end{cases} \tag{13}$$

Structure of Encoder and Decoder

A code word from $A(g,h)$ codeset is assigned to each user, where the number of supported user in the system is N_X N_Y. The transmitter encode the data as per the code word $A(_{g,h})$ assigned and transmit the data to all the receivers. The receiver recover the data by correlating with the assigned code word $A(g_{,h})$.

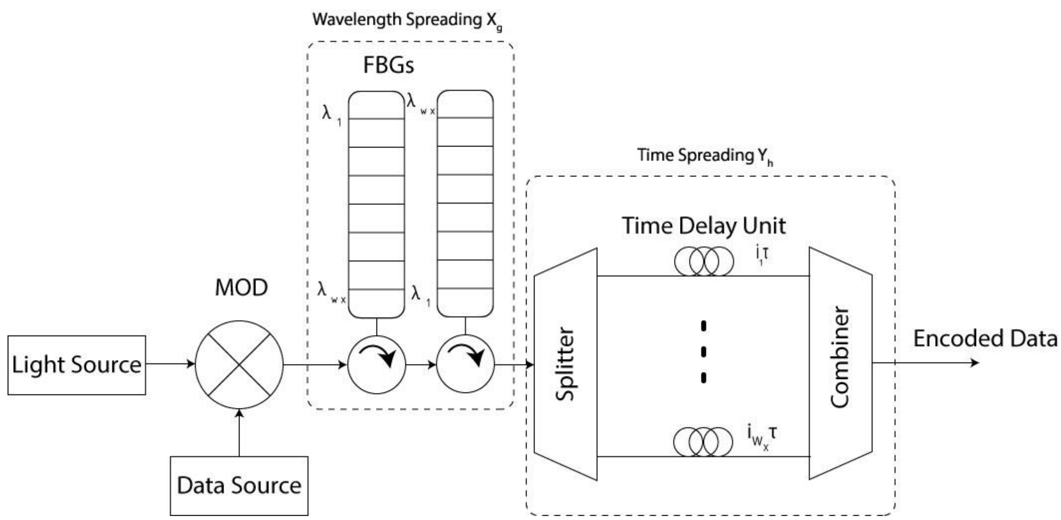

Figure 1: Structure of Encoder for EMD/MD code

A transmitter corresponding to $A_{(g,h)}$ consist of a broadband light source, a data modulator, $1\times\omega_X$ optical splitter, ω_X optical Basel filters, $\omega_X \times 1$ optical combiner, and ω_Y time delay units. At the transmitter, the broadband light is modulated with data stream using ON-OFF keying such that optical pulse is transmitted if the data bit is high only. This modulated broadband light is split into ω_x spectral tuples. Spectral tuple is transmitted according to the code sequence X_g by using optical Basel filter. Optical Basel filter only allows specific spectral components and block the reset of the light spectrum and the allowed ω_X spectral components are combined using $1\times \omega_X$ optical combiner.

The encoded light containing ω_X spectral tuple is then passed through time delay unit for temporal encoding. A, $1\times\omega_Y$ optical splitter split the spectral encoded light into ω_Y and each tuple is delayed by $i\tau$ according to temporal dimension Y_h of the code word $A_{(g,h)}$. The delay of each line is proportional to the position of the bit in the sequence Y_h. After introducing the delay, the tuples are combined using $\omega_{Y\times 1}$ optical combiner and finally the output of the encoder is encoded by wavelength and time according to the code word $A_{(g,h)}$ [23].

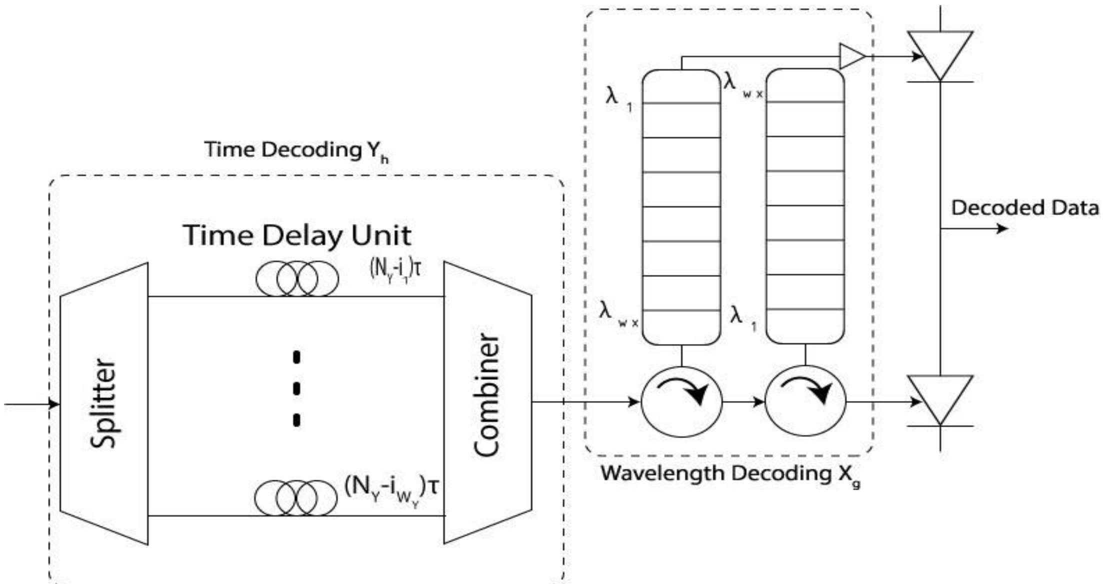

Figure 2: Structure of Decoder for 2D EMD/MD code

We used MATLAB R2019b to train and test our suggested ANN model. To train the system, the Levenberg-Marquardt training algorithm employs a three-layer feedforward backpropagation neural network, with input, hidden, and output layers serving as the essential layers. Several neuron counts in the hidden layer were investigated during the training phase, and 35 neurons yielded the best results. The output layer only has one neuron for load prediction output. The input-output links are built using learning algorithms from the data itself, with weights changing at each iteration based on error reduction [22].

Results and Discussion

Simulation Setup for Transmitter

For each user, the transmitter consists of broadband light starting from 1490 nm Light Emitted Diode (LED) driven by a DC source. The broadband light of LED is split into ω_X tuples using optical splitter. The $\omega_X\times 1$ WDM Mux uses Basel filter of second order with each tuple spectral width of 0.4 nm. The selection of tuples are based on the X_g sequence for each user A_(g,h). The spectrally encoded signal is then modulated with data using Mach-Zehnher Modulator (MZM). A random data is generated form Pseudo Random Bit Sequences (PRBS) generator and then converted to electrical pulses using Non-Return to Zero (NRZ) pulses generator. The MZM modulator modulates optical signal with this data pulses using On-Off Keying.

Back-propagation learning involved transmitting inputs from one layer to the next until calculated outcomes were created and compared to real output to compute the error. The weights and biases were changed layer by layer by propagating mistakes back into the input. The fundamental backpropagation learning algorithm is the steepest descent algorithm that can minimize the sum of squares of mistakes. This approach, however, converges slowly and is not numerically efficient. Momentum and learning rate are two factors that can accelerate the algorithm process. The learning rate is defined as the proportion of the error gradient that controls the weights. Fast convergence happens at higher levels, although oscillations become more intense. The momentum defines the proportion of earlier weight changes that are taken into account when computing new weights [22].

DATA-Set Selection

We obtained raw hourly load data from the ISO New England Pool region from January 1st, 2017 to June 30, 2021. After that, the yearly load data is put into a single data set of 39,408 data points. The data collection includes the date, hour, dry-bulb (°F), dew-point (°F), and system electrical load (MW). We selected 24 data points since we used hourly load data and there are 24 hours in a day. MATLAB software is used to import datasets, generate ANN models with specified inputs, and finally forecast and show the tested results to determine how well the intended model performs. Another set of holiday data from 2017 to 2021 is utilized as input in our dataset.

Load Analysis

The ANN model is system-dependent, and as a key step forward in developing ANN for momentary load prediction, the necessary framework attributes should be investigated. The initial step in building any load predictor is to examine historical load records to extract load features such as periodicity and trends.

Following preprocessing, the dataset was carefully examined to observe the variation in load versus different hours of the day. The table below shows the real monthly peak load numbers from 2017 through June 2021, along with the day and hour of the day.

Table 1. Monthly basis Peak Load

Year/Month	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
Day/ Hour 2017	9/18	9/19	15/20	6/18	18/8	13/17	19/18	22/17	27/17	9/19	28/18	28/18
2017 Load (MW)	19592	18165	17502	15843	20250	23968	23579	22769	20999	17255	17079	20524
Day/ Hour 2018	5/18	7/18	7/19	3/20	29/18	18/17	5/18	29/17	6/16	10/19	15/18	18/18
2018 Load (MW)	20662	18308	16943	15778	17518	21076	24512	26024	24475	17479	17590	18466
Day/ Hour 2019	21/18	1/19	6/19	9/20	20/18	28/18	30/18	19/16	23/17	2/15	13/18	19/18
2019 Load (MW)	20773	18585	17876	15034	15748	19913	24361	23365	19162	16138	17548	19065
Day/ Hour 2020	20/18	14/19	1/19	27/18	29/18	23/18	27/18	11/18	10/18	30/19	18/18	17/18
2020 Load (MW)	18097	16991	15888	14254	16593	21519	25121	24335	19260	15616	17157	18922
Day/ Hour 2021	29/18	1/18	2/19	16/12	26/18	29/16	--------	--------	--------	--------	--------	--------
2021 Load (MW)	18839	18185	17738	14649	18846	25726	--------	--------	--------	--------	--------	--------

According to the statistics, the greatest load value is 26024 MW for all the years, which happened in the seventeenth hour on 29th August 2018. Seventeenth, Eighteenth, and Nineteenth hours of the day have the highest peak load. Load increases throughout the summer months (June-September), whereas it decreases during the winter months (October-May). The MATLAB annual load graphs shown below will assist you in interpreting lines in the preceding:

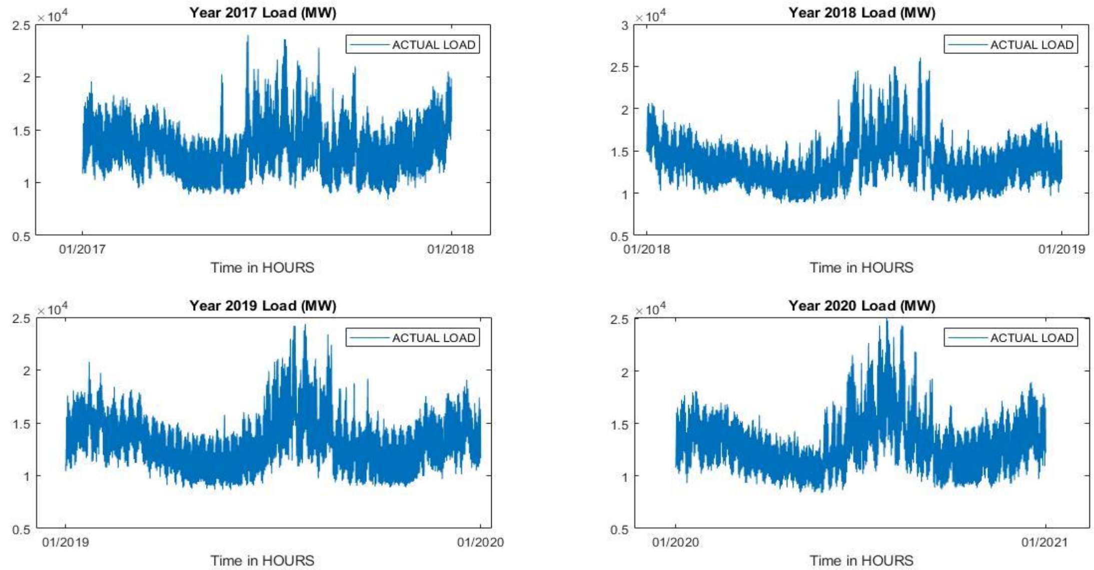

Fig. 2. Yearly Actual load data

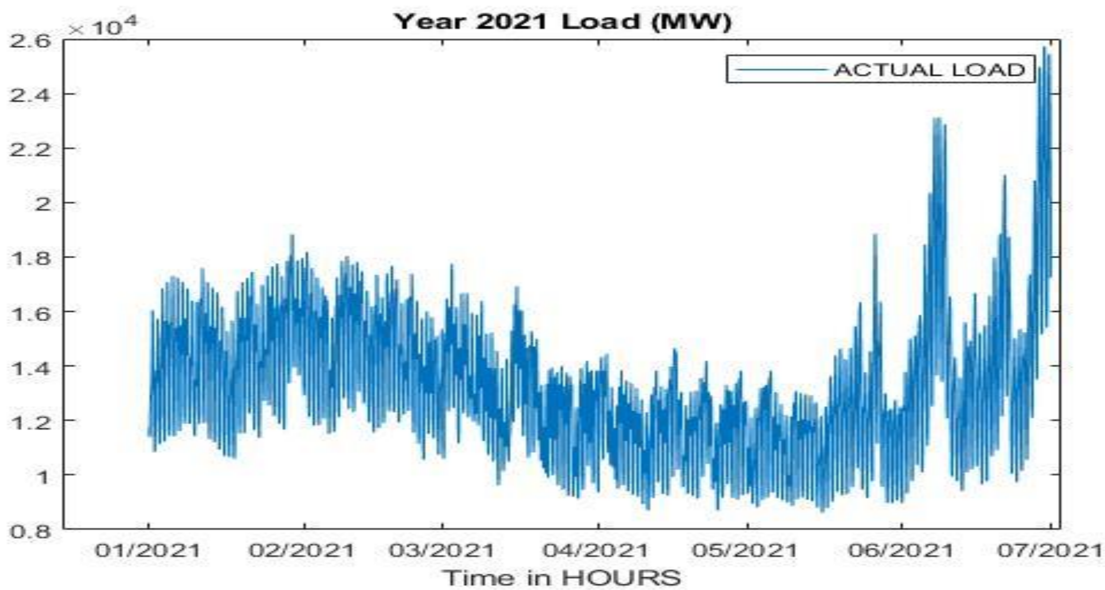

Fig. 3. 2021 forecast actual load data.

In the Figure 3, demonstrates intermittent power surges and unexpected load drops throughout the year, particularly in the summer. However, overall load changes are essentially the same over time.

Data Analysis

When examining load trends over the course of a year, it is critical to account for critical factors such as weather conditions and public holidays. Previous study has shown that temperature and humidity have a significant impact on the dynamic variations in electrical load behaviour. While wind speed, air pressure, weather patterns, geographical location, public interruptions, and lockdowns can all have an effect on load behaviour, they were not particularly explored in this study. Table 2 depicts the actual load fluctuations for each hour on June 30, 2021, illuminating the link with dew point and dry bulb readings.

Table 2: Dew-Point, Dry-Bulb, & Actual Load Data

Hour / Weather Information June 30, 2021	Dry Bulb (ºF)	Dew Point (ºF)	Actual Load (MW)
1	78	71	17,517
2	77	71	16,549
3	77	70	15,881
4	77	71	15,459
5	76	70	15,416
6	75	70	15,830
7	75	69	17,095
8	78	70	18,756
9	80	70	20,143
10	83	70	21,233
11	86	70	22,299
12	89	70	23,246
13	91	70	24,087
14	92	70	24,879
15	93	69	25,333
16	94	68	25,420
17	94	69	25,436
18	91	69	25,153
19	86	69	24,020
20	78	70	22,980
21	75	70	21,888
22	75	70	20,481
23	74	70	18,854
24	73	70	17,249

As shown in Table 2, on the load there is a little dew point effect, with the dry bulb temperature (oF) appearing as an important component trendy load variation. There is a direct association seen; as the dry bulb value increases, so does the load, and vice versa, as the dry bulb value decreases, so does the load. Figure 4 depicts graphs covering the whole dataset of dew point, dry bulb, and real load demand to present a holistic perspective, encouraging deeper examination for more insights into their interplay.

Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 6, 2025

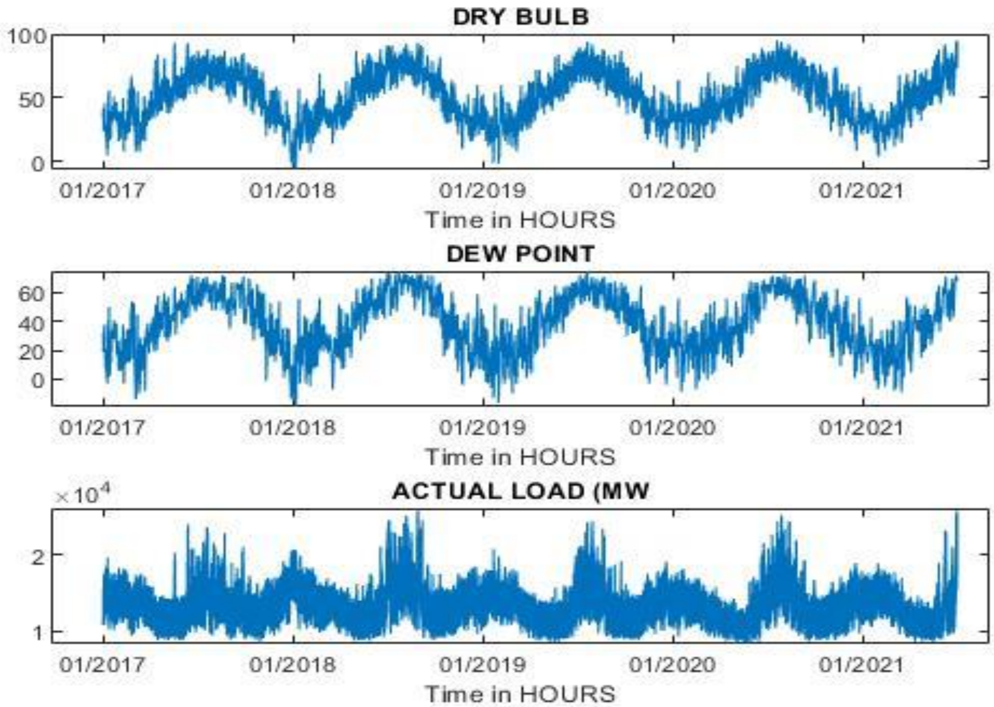

Fig. 4. Load Analysis, Dew-point, and Dry-bulb

Datasets Distribution

Datasets were divided into two broad categories: Testing Data and Training Data. In 80% of situations, data was picked for training and 20% for testing. To get the necessary weights for the ANN model, the Levenberg-Marquardt back-propagation technique was utilised. Finally, the dataset was put through its paces. After multiple trial-and-error tries with varied numbers of neurons, the precise timely network stayed chosen based on the least MAPE criteria. Hidden-layer sigmoid-transfer functions were used to evaluate the models on 20-40 neurons. 1-5 hidden layers were utilised in a trial-and-error fashion, with 35 neurons, one hidden layer, and one output layer producing the best results.

For testing purposes the month of June 2021 was picked in the second stage of the STLF forecasting procedure, then the outstanding whole previous dataset data was estimated during model exercise. During the last week of June 2021, the expected load values were tested in the third phase. The 30th of June 2021 was chosen as the last day of the fourth phase of power load forecasting. In the last stage, we forecasted the last hour of June 30th, 2021.

In this study, we concentrated on weekly and daily load data as the inputs and goal data were same throughout all stages, resulting in nearly identical outcomes.

Simulation Results

Before developing Artificial Neural Networks (ANN), a preliminary analysis was carried out with the Multiple Linear Regression (MLR) approach to determine the predicted load values for all 39,408 data points in the dataset. A comparison of the actual electrical demands (MW) and the corresponding projections produced by the MLR technique is shown in Figure 5. Notably, other inputs were not included in the study when the MLR technique was evaluated; only regular load data was taken into account.

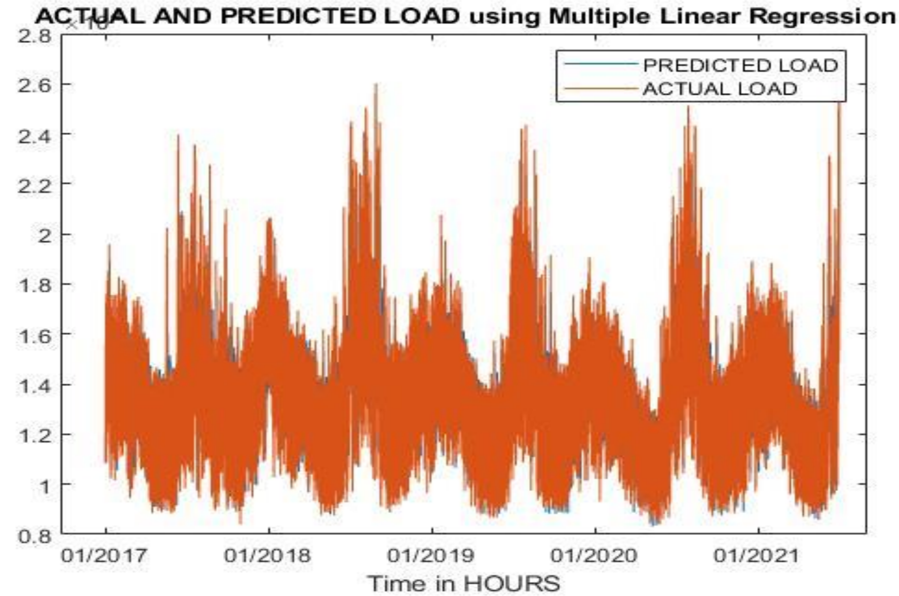

Fig. 5. Predicted & actual load

When it is zoomed its results are shown more clearly as:

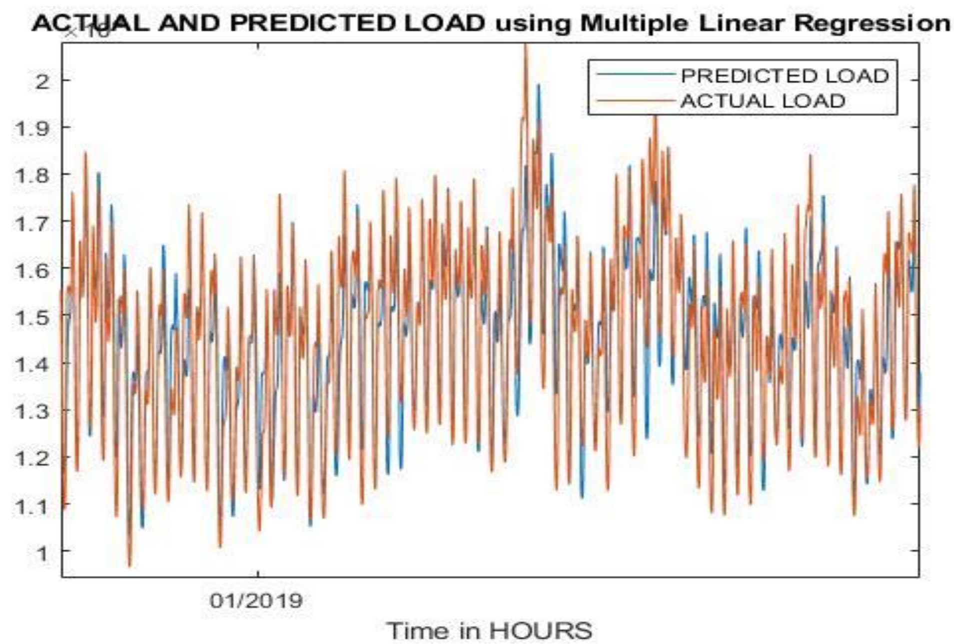

Fig. 6. Clear Plot of MLR

Findings indicates the real and expected load statistics differ significantly. Only a few deviations occur along the linear variable line, with the majority occurring under peak loads. As an outcome, the approach is ineffective for adjustable loads then effective for linearly changing loads.

The MATLAB R2019b Deep Learning Toolbox was used to simulate the FITNET Artificial Neural Network (ANN) models. Several data sets were simulated in a feedforward network using the same dataset, and the results were methodically recorded for various neuron pairings and repetitions. In addition to applying the Levenberg-Marquardt (L-M) training procedure, the networks were also examined for Bayesian Regularisation (BR). Nevertheless, the BR approach performed noticeably worse than L-M, which resulted in its removal from the report and the conclusions drawn from it not being taken into account in the end.

Arithmetical features of the hourly, weekly, and monthly load data are shown in Figure 7, which also shows the load changes that may be observed, such as weekend decreases, month-to-month variations, and hourly variations. Since these unique load behaviour patterns are determined by network parameters, they must be examined in order to be considered when creating an appropriate Artificial Neural Network (ANN) model [23].

During the training phase, all other parameters (training procedures, transfer functions, hidden layers, and performance functions) were held constant, but the number of neurons was regularly changed between 20 and 40. The results of this variation—which concentrate on the hourly load projections—are shown in the table. A detailed visual comparison of the actual load and the accompanying forecasts throughout a 24-hour period is shown in Figure 7. It is noteworthy that the trough occurred between 3 and 4 hours, while the highest load was recorded between 17 and 18 hours. This research provides insights into the temporal dynamics of load forecasts and illuminates how sensitive the model is to variations in the number of neurons.

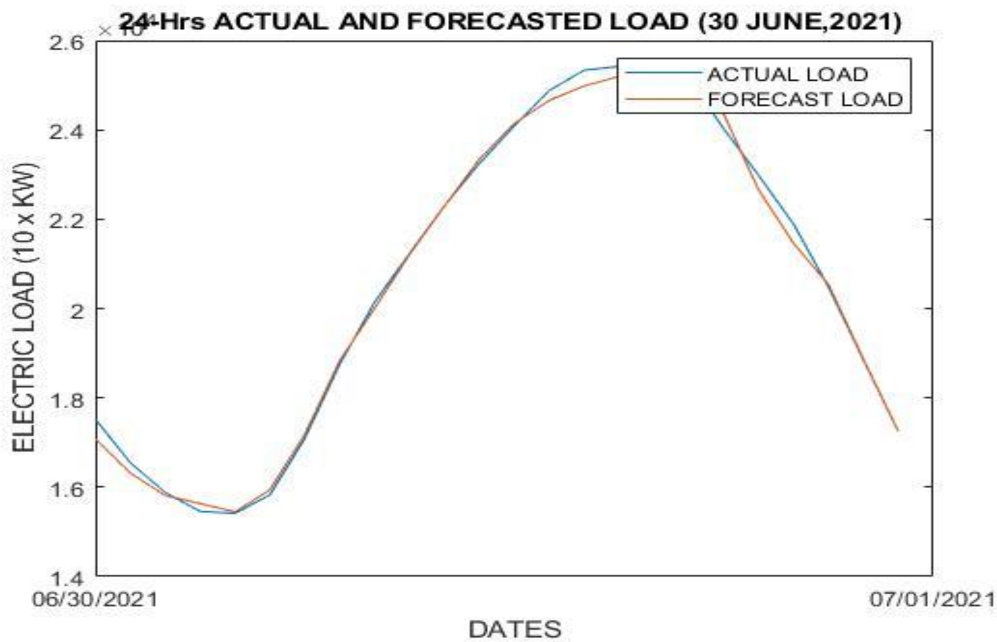

Fig. 7. 24 hours forecast & actual load

The graph showing the real and expected load shows that there are very little fluctuations throughout peak and midday hours. The graph's stability highlights how well our Artificial Neural Network (ANN) model design predicts similar load data. The model's aptitude for precise load forecasts is confirmed by the consistency in performance across peak and noon situations, which demonstrate the model's resilience and dependability in capturing the underlying patterns and trends in the load data.

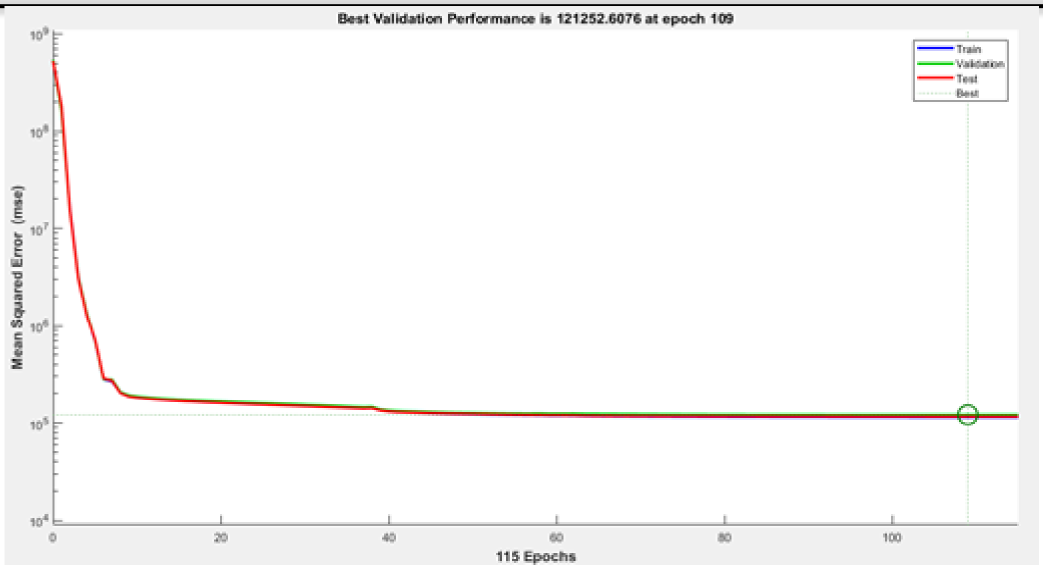

Fig. 8. Number of EPPCH's Against Mean Squared Error

Using the Levenberg-Marquardt back-propagation technique produced convergence after 115 iterations and 109 epochs. The results show stability post-convergence, with no appreciable increase, as seen in Figure 8. Refinement is evident in the output, which is characterised by a decrease in data loss and an increase in accuracy. There isn't any divergence in mistakes when looking at the convergence charts for the training set.

In Figures 10 and 11, the regression plot depicting the target values against the predicted load data, and the comprehensive regression plots encompassing training, validation, testing, and overall datasets. The assessment of targeted and projected errors employs a regression measure, with Figure 12 illustrating the histogram of fit-set errors. Figure 13 provides a visual representation of both the distribution of absolute errors and the distribution of absolute percent errors, allowing for a thorough comparison of performance.

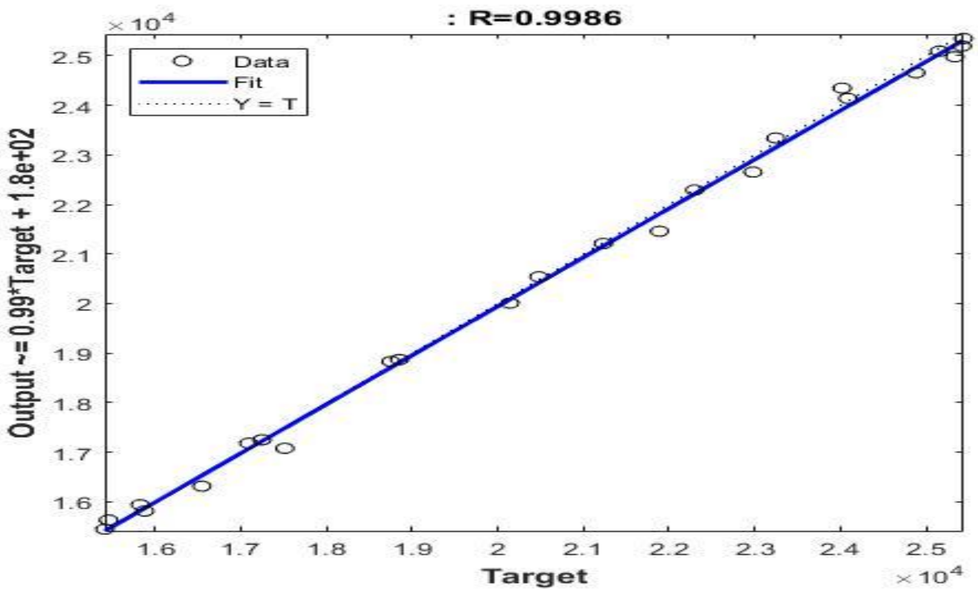

Fig. 9. Neural Network Training Curve of Performance

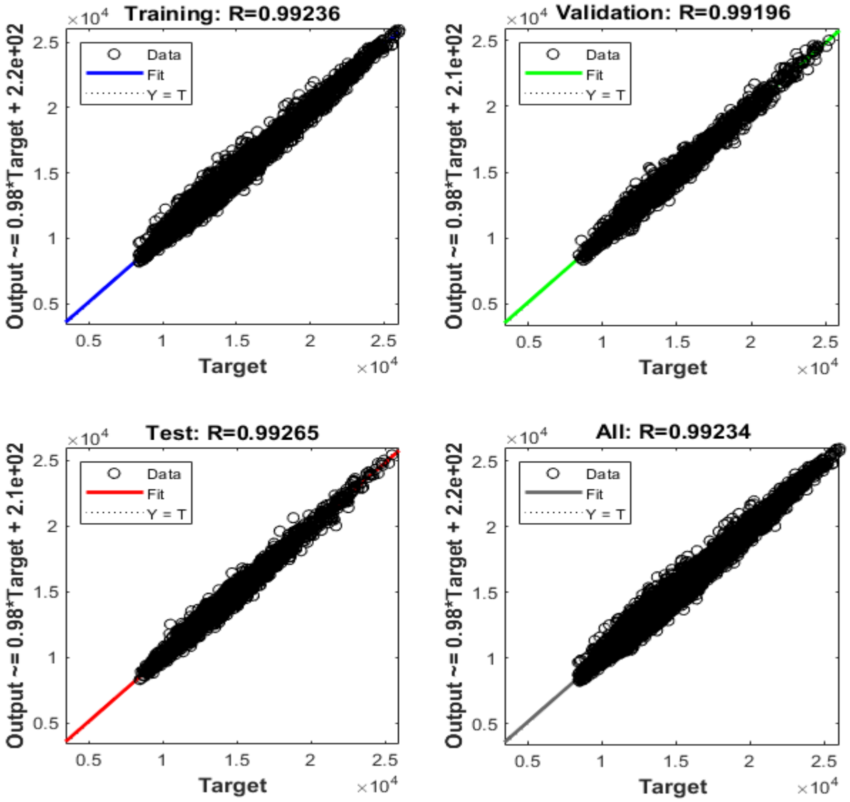

Fig. 10. ANN Complete Regression Plot

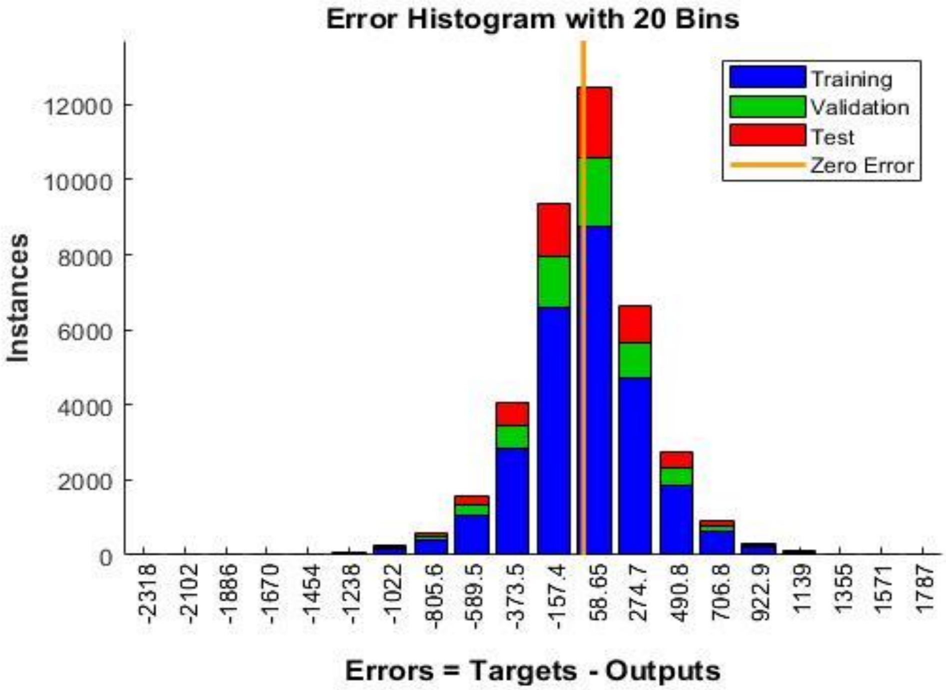

Fig. 11. Histogram Error

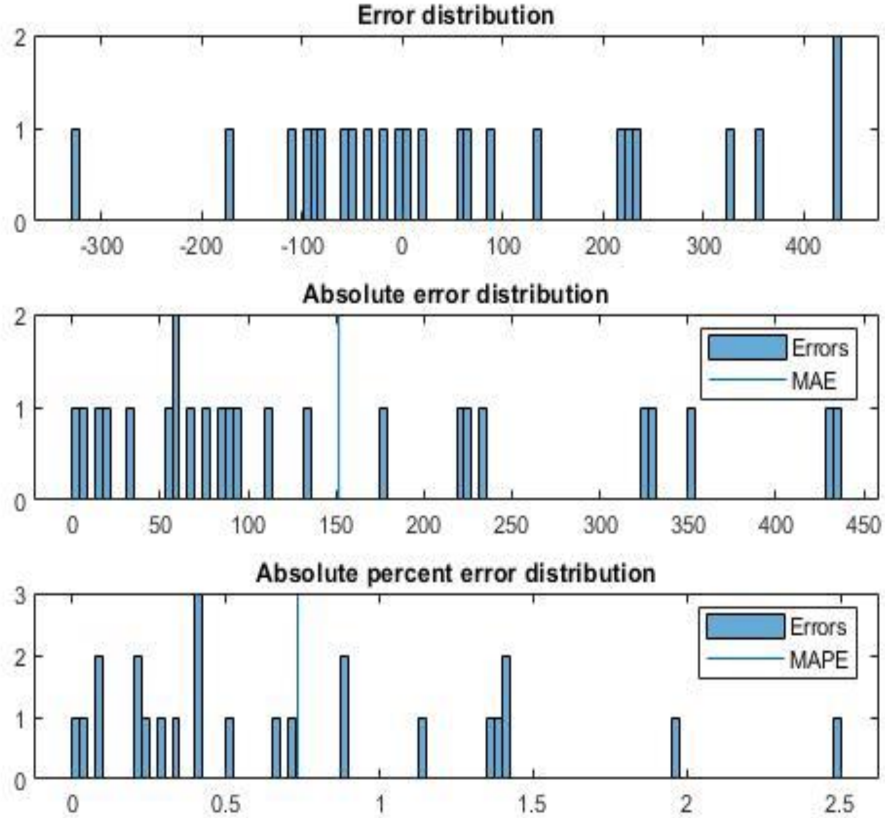

Fig. 12. Distribution Error

Table 3 depicts the projected 24-hour load data for June 30th, 2021 for each example used to estimate effectiveness by varying the number of neurons.

Table 3. Actual and Predicted Loads of 24-Hours

Hr's	PL For n = 40	PL For n = 35	PL For n = 30	PL For n = 25	PL For n = 20	AL (MW)
1st Hr.	17226	17108	16853	17000	17014.5	17,517
2nd Hr.	16393	16407	16054	16379	16301	16,549
3rd Hr.	15970	15935	15591	16026	15892.75	15,881
4th Hr.	15915	15764	15448	15902	15821	15,459
5th Hr.	15772	15525	15316	15678	15573	15,416
6th Hr.	16041	15822	15705	15950	15765.5	15,830
7th Hr.	16883	16787	16908	16977	16780.6	17,095
8th Hr.	18640	18730	18942	18534	18765.5	18,756
9th Hr.	19779	20071	20108	19793	19848	20,143
10th Hr.	20865	21182	21129	20994	20916.7	21,233
11th Hr.	22011	22207	22108	22261	22013	22,299
12th Hr.	23305	23328	23180	23481	23095	23,246

13th Hr.	24468	24322	24168	24294	23938.6	24,087
14th Hr.	25342	24964	24948	24741	24580	24,879
15th Hr.	25986	25261	25417	25028	25113	25,333
16th Hr.	26505	25442	25623	25365	25547.9	25,420
17th Hr.	26808	25626	25759	25825	25871	25,436
18th Hr.	26089	25368	25393	25460	25576	25,153
19th Hr.	24669	24623	24579	24166	24791	24,020
20th Hr.	22464	22837	22993	22026	22881	22,980
21st Hr.	21121	21483	21850	20841	21465	21,888
22nd Hr.	20213	20581	20893	20069	20551.8	20,481
23rd Hr.	18752	18972	19280	18879	18891	18,854
24th Hr.	16808	17104	17418	17291	16924.6	17,249

Table 4: Predicted Load Performance

	n = 40	n = 35	n = 30	n = 25	n = 20
Number of Iterations	53	115	164	425	281
Regression	0.99222	0.9986	0.99788	0.99458	0.99510
EPOCH	47	109	158	419	274
Performance	**1.1826e+05**	**1.1005e+05**	**1.1219e+05**	**1.2761e+05**	**1.1814e+05**

In Table 4, the findings of a huge dataset including 39,385 training data points compared to a simple 24 data points during a 24-hour testing period, using hourly data received from June 30, 2021. The neural network model was rigorously tested under a variety of scenarios, each corresponding to a different number of neurons chosen, with the ultimate objective of determining the best data aggregation. When 35 neurons are used, the MAPE error is 0.73 percent, and it rises to 0.93 percent when 40 neurons are used. During the neural network training with 35 neurons, the peak performance curve reached 0.9986. As a consequence, when confronted with 24-hour test data, the best load forecasting outcomes were obtained by deploying 35 neurons within a single hidden layer, as demonstrated by the acquired findings.

The use of 30 neurons resulted in optimal weekly load analysis findings with notable accuracy. Notably, the training regression value is 0.99213, R=0.99215 summarises the total response, whereas the testing regression score is 0.99211 and the validation regression score is 0.99229. The best fit peaks at 220 epochs and 226 iterations with a strong regression value of R=0.99856, excellent performance shown by P=1.1171e+05, and a low MAPE of 1.40 percent. Figure 13 depicts the graphical display of error statistics for both weekly and hourly load data, emphasising the rigorous comparison between actual and expected loads during the final week of June 2021.

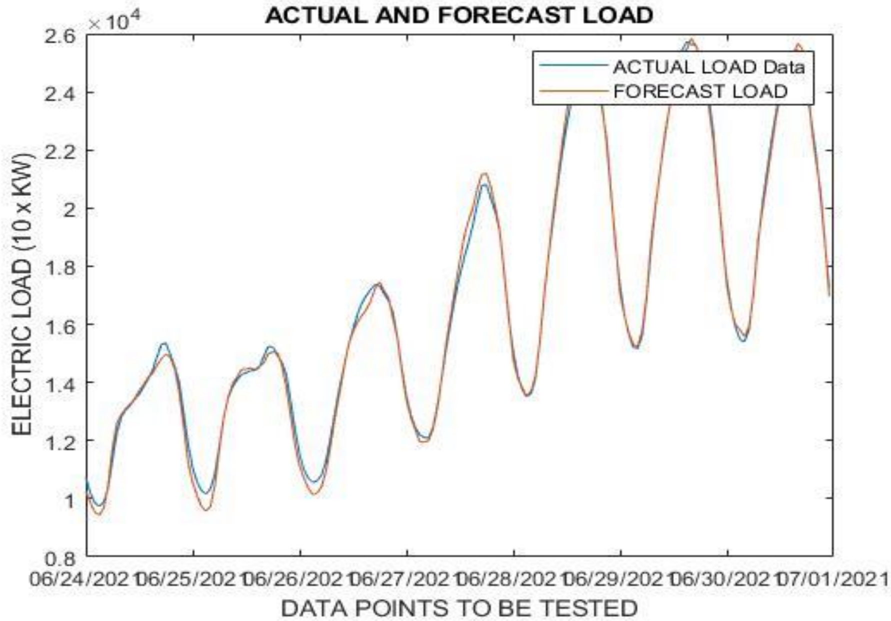

Fig. 13. Actual and Forecast Load & Actual Data (June2021)

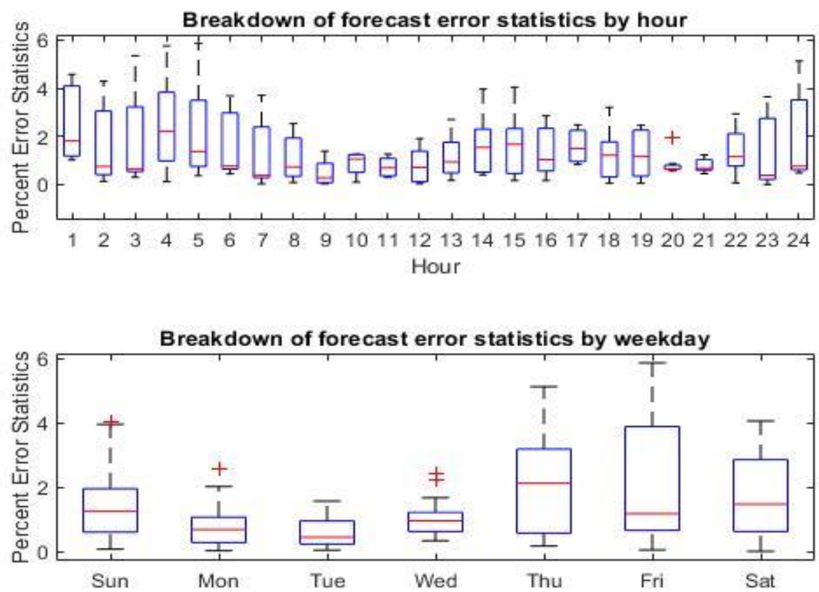

Fig. 14. Time & Electric Load Correlation

Figure 14 highlights diverse load ranges by illustrating load distribution over various hours and days of the week. Notably, Sundays have the lowest load levels during the week, indicating a trough in the load pattern. Fridays, on the other hand, have the greatest load levels, forming the apex of the weekly load distribution. This graphic depiction captures the variations in load intensity across different days and hours, providing insights into the load profile's dynamic character throughout the week.The expected hourly load curves follow the same pattern as the weekly load curves. There are tiny oscillations at the load peaks, and no aberrations in the load lines change linearly. To decrease these slight changes, the model will need to be fine-tuned by modifying the weights/biases and other parameters that were kept constant during our network training.

Conclusions

A Novel Feedforward (FITNET) Neural Network implementation for short-term load prediction (STLF) was simulated in MATLAB software, and the technique indicates that ANN models may be prepared for training by employing numerous types and sequences of real-time inputs. The data was collected over a four-and-a-half-year period in the ISO New-England NE-Pool region and organised into a single data collection. Time and climatic factors (time, dew point, dry bulb), as well as weekdays, were employed as primary inputs (output data). For different orders of neurons, the ANN model was trained using the Levenberg-Marquardt backpropagation method. To achieve the best results, the model was tested numerous times for weekly and daily load forecast approaches. The ANN model's network efficiency was enhanced by attaining 0.73% MAPE error for hourly load prediction and 1.40% MAPE error for weekly load forecasting, which are both highly acceptable.

REFERENCES

[1] T. K. Chheepa, and T. Manglani, "A Critical review on employed techniques for short term load forecasting", International Research Journal of Engineering and Technology, vol. 4, No. 6, June 2017.

[2] R. Hu, S. Wen, Z. Zeng, and T. Huang, "A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm", Neurocomputing, vol. 221, pp. 24-31, 2017. https://doi.org/10.1016/j.neucom.2016.09.027

[3] S. Khatoon, and A.K. Singh, "Analysis and comparison of various methods available for load forecasting: An overview", In 2014 Innovative Applications of Computational Intelligence on Power, Energy and Controls with their impact on Humanity (CIPECH), pp. 243-247, Nov. 2014. https://doi.org/10.1109/CIPECH.2014.7019112

[4] K. H. Baesmat, and A. Shiri, "A new combined method for future energy forecasting in electrical networks", International Transactions on Electrical Energy Systems, vol. 29, no. 3, pp. 2749, 2019. https://doi.org/10.1002/etep.2749

[5] K. G. Boroojeni, M. H. Amini, S. Bahrami, S. S. Iyengar, A. I. Sarwat, and O. Karabasoglu, "A novel multi-time-scale modeling for electric power demand forecasting: From short-term to medium-term horizon", Electric Power Systems Research, vol. 142, pp. 58-73, 2017. https://doi.org/10.1016/j.epsr.2016.08.031

[6] K. Zor, O. Timur, and A. Teke, "A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting", In2017 6th International Youth Conference on Energy (IYCE), pp. 1-7, June 2017. https://doi.org/10.1109/IYCE.2017.8003734

[7] L. Suganthi, and A. A. Samuel, "Energy models for demand forecasting-A review", Renewable and sustainable energy reviews, vol. 16, no. 2, pp. 1223-1240, 2012. https://doi.org/10.1016/j.rser.2011.08.014

[8] S. Ryu, J. Noh, and H. Kim, "Deep neural network-based demand-side short term load forecasting", Energies, vol. 10, no. 1, pp. 3, 2017. https://doi.org/10.3390/en10010003

[9] K. B. Debnath, and M. Mourshed, "Forecasting methods in energy planning models", Renewable and Sustainable Energy Reviews, vol. 88, pp. 297-325, 2018. https://doi.org/10.1016/j.rser.2018.02.002

[10] F. Rodrigues, C. Caldeira, and J. M. F. Calado, "Neural networks applied to short term load forecasting: A case study", In Smart Energy Control Systems for Sustainable Buildings, pp. 173-197, 2017. https://doi.org/10.1007/978-3-319-52076-6_8

[11] M. Mordjaoui, S. Haddad, A. Medoued, and A. Laouafi, "Electric load forecasting by using dynamic neural network", International journal of hydrogen energy, vol. 42, no. 28, pp. 17655-17663, 2017. https://doi.org/10.1016/j.ijhydene.2017.03.101

[12] S. Matthew, and S. Satyanarayana, "An overview of short-term load forecasting in electrical power system using fuzzy controller", In 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 296-300, Sep. 2016. https://doi.org/10.1109/ICRITO.2016.7784969

[13] D. K. Chaturvedi, and S. A. Premdayal, "Short Term Load Forecasting (STLF) Using Generalized Neural Network (GNN) Trained with Adaptive GA", In International Conference on Swarm, Evolutionary, and Memetic Computing, pp. 132-143, Dec. 2013. https://doi.org/10.1007/978-3-319-03756-1_12

[14] Haque, S. A., & Islam, M. A. (2021). Artificial Neural Network-Based Short-Term Load Forecasting for Mymensingh Area of Bangladesh. International Journal of Electrical and Computer Engineering, 15(3), 99-103.

[15] Anand, A., & Suganthi, L. (2018). Hybrid GA-PSO optimization of artificial neural network for forecasting electricity demand. Energies, 11(4), 728. https://doi.org/10.3390/en11040728

[16] Shafiei Chafi, Z., & Afrakhte, H. (2021). Short-Term Load Forecasting Using Neural Network and Particle Swarm Optimization (PSO) Algorithm. Mathematical Problems in Engineering, 2021. https://doi.org/10.1155/2021/5598267

[17] Haque, S. A., & Islam, M. A. (2021). Artificial Neural Network-Based Short-Term Load Forecasting for Mymensingh Area of Bangladesh. International Journal of Electrical and Computer Engineering, 15(3), 99-103.

[18] Zheng, X., Ran, X., & Cai, M. (2020). Short-term load forecasting of power system based on neural network intelligent algorithm. IEEE Access. https://doi.org/10.1109/ACCESS.2020.3021064

[19] Mordjaoui, M., Haddad, S., Medoued, A., & Laouafi, A. (2017). Electric load forecasting by using dynamic neural network. International journal of hydrogen energy, 42(28), 17655-17663. https://doi.org/10.1016/j.ijhydene.2017.03.101

[20] Mohammad, S., & Hasan, M. K. An Effective Artificial Neural Network based Power Load Prediction Algorithm. International Journal of Computer Applications, 975, 8887.

[21] Nguyen, T. A., Ly, H. B., Mai, H. V. T., & Tran, V. Q. (2021). On the Training Algorithms for Artificial Neural Network in Predicting the Shear Strength of Deep Beams. Complexity, 2021. https://doi.org/10.1155/2021/5548988

[22] Rodrigues, F., Cardeira, C., & Calado, J. M. F. (2017). Neural networks applied to short term load forecasting: A case study. In Smart Energy Control Systems for Sustainable Buildings (pp. 173-197). Springer, Cham. https://doi.org/10.1007/978-3-319-52076-6_8

[23] Buitrago, J., & Asfour, S. (2017). Short-term forecasting of electric loads using nonlinear autoregressive artificial neural networks with exogenous vector inputs. Energies, 10(1), 40. https://doi.org/10.3390/en10010040