Load Forecasting on Edge: A Step Towards Decentralized Energy Management with Data Compression and Encryption

Ali Oveysikian[1,*]
[1] *PhD student, Department of computer science and engineering, tarbiat modares university, Tehran, Iran*
ali.oveysikian@modares.ac.ir

ABSTRACT

In the modern age of swift technological advancement, smart meters have become a fundamental element of smart grid infrastructures, serving a critical role in acquiring energy data and delivering advanced services. These devices provide essential insights into energy consumption patterns, enabling more efficient energy management, cost reduction, and enhanced grid stability. Among the various applications of smart meter data, short-term load forecasting (STLF) for smart homes is notably crucial. However, the increasing deployment of smart meters and the vast amounts of data they generate pose considerable challenges regarding to storage, processing, transmission, and data security. To address these issues, edge computing, data compression, and encryption have emerged as transformative technologies, offered practical solutions and enhancing the effectiveness of load forecasting systems.

This paper presents a novel framework leveraging Multi-Layer Edge Computing (MLEC) for load forecasting, data compression, and encryption to address the challenges of decentralized energy management. By processing energy data locally, the framework enhances real-time decision-making, minimizes dependency on cloud platforms, and secures user privacy. The proposed framework optimizes and simplifies short-term load forecasting by achieving high accuracy while maintaining computational simplicity across various scenarios. It enhances data management through efficient compression, balancing compression ratios with processing speed and energy efficiency, and ensures secure data transmission with minimal overhead via secondary encryption. Simulations demonstrate prediction accuracy within an acceptable range, a 20-25% reduction in data size, and low-latency performance, highlighting the framework's scalability and effectiveness for decentralized smart grids.

Keywords: *Load Forecasting, Multi-Layer Edge Computing, Data Compression, Encryption, Internet of things, Smart Meter*

1. *INTRODUCTION*

In recent years, driven by population growth, advancements in digital technologies, and the emergence of Industry 4.0, the rapid growth in energy demand has highlighted the need for efficient management and the transformation of traditional energy infrastructures into smart grids. As one of the key elements of smart grids, smart meters serve as the backbone of communication, facilitating two-way interactions between end-users and power systems. These devices consistently track energy consumption over various time horizons, such as real-time and semi-real-time, and transmit the collected data to the relevant servers. This data is a critical input for various applications, including load forecasting, dynamic pricing strategies, identifying unauthorized consumption, detecting energy theft, customer segmentation, and providing personalized energy services.

Over the past decade, the integration of smart meters with modern power systems has revolutionized energy management and paved the way for the carrying out initiatives such as active consumers. However, the exponential growth in the volume of data generated by smart meters has introduced significant challenges.

These challenges can be categorized into issues related to the required infrastructure for data storage, processing, and transmission. While the vast amount of data is crucial for improving the accuracy of studies and facilitating better decision-making, it also significantly increases the risks of cybersecurity threats and privacy breaches. Ensuring the secure transmission and storage of this data, while enabling precise and efficient analysis, has become a top priority for the effective utilization of smart grid infrastructures. To balance the need for precise data analysis with robust cybersecurity, many solutions such as edge computing, data compression, and strong encryption protocols have been proposed. These advancements are important to addressing the dual challenge of managing the vast volume of energy data and safeguarding user confidentiality in the era of digital energy systems. Accurate load forecasting(LF), particularly for short-term periods(from minutes to hours), is an important factor in enhancing power grid operation. Such forecasts enable grid operators to precisely determine the exact required amount of energy to maintain a balance between supply and demand and control the frequency of power grid. Preventing peak load conditions, which can lead to regional blackouts, is another benefit of accurate forecasting. Additionally, these processes reduce unnecessary costs associated with overproduction or the emergency use of storage resources, thus enhancing the grid's economic efficiency. However, achieving high levels of accuracy in energy demand forecasting poses several challenges.

All existing approaches, like machine learning [1], [2], [3] , *deep learning* [4], [5] *and time-series* [6], [7] *models, require substantial amounts of data and computational resources. The processing and analysis of such data often needs significant processing power and memory. Furthermore, while cloud-based platforms offer high computational capabilities, they are often costly and involve longer times for data transmission and computation. These limitations make it difficult to meet the real-time requirements of grid operators for short-term energy demand predictions. To address these challenges, edge computing has emerged as an efficient alternative to cloud-based models. By processing data closer to the source, edge computing reduces processing times and minimizes the costs of data transmission and cloud resources. Combined with data compression and encryption algorithms, this technology can provide a comprehensive solution for real-time, secure energy demand forecasting.*

This paper proposes a novel framework based on Multi-Layer Edge Computing (MLEC) to address these challenges. The proposed framework not only reduces data volume and enhances the security of data transmission but also enables complete load forecasting computations at the edge level. This approach minimizes dependency on cloud platforms, facilitates real-time data analysis, and effectively preserves user privacy. One of the key applications of this framework lies in energy management for smart homes. In smart homes, accurate load forecasting allows system operators to allocate energy resources more efficiently. For instance, STLF enables the scheduling of household appliances during periods of low electricity costs, preventing load fluctuations during peak hours. This capability significantly reduces household energy expenses and improves the overall efficiency of the power grid. Additionally, by reducing the volume of data transmitted to central servers and securely encrypting it, the security and resilience of smart home systems are enhanced. Furthermore, by processing data locally and minimizing latency, this framework enables real-time execution of energy monitoring and control systems. Moreover, the use of MLEC framework facilitates the analysis of individual consumption patterns, allowing for personalized services, such as recommending optimal times for operating high-energy appliances.

The rest of this paper is organized as follows: Section 2, focuses on the topic of load forecasting at the edge, providing a detailed review of related studies and highlighting advancements and challenges in this field. Section 3, delves into data compression techniques, categorizing various methods for minimizing data volumes generated by smart meters. This section emphasizes the importance of data compression in optimizing bandwidth utilization and reducing storage costs. In Section 4, encryption methods are explored, offering a comprehensive review of recent developments and their applicability to smart meter data. Special attention is paid to practical encryption algorithms tailored for securing data in smart grid environments. Section 5, examines the role and methodologies of load forecasting, providing insights into its critical applications in energy management. Section 6, introduces the proposed framework in detail, discussing its architecture and the hardware components. This section also presents the results of the framework's implementation, evaluating its performance in terms of prediction accuracy, data compression rates, and energy consumption. In conclusion section, this study summarizes the findings, highlighting the proposed framework's exceptional performance in prediction accuracy, data security, energy efficiency, and operational cost-effectiveness. These achievements position the framework as a highly effective and scalable solution for the evolving needs of future

smart grids. Additionally, the study outlines potential directions for future research to further enhance the framework's capabilities and adaptability.

2. *RELATED WORK*

In recent years, load forecasting has become a critical area in energy management and power grid. numerous research efforts have been dedicated to this topic. These studies can be categorized based on various perspectives, such as forecasting horizon (short-term, mid-term, or long-term), forecasting level (grid-level, regional, or household), consumer type (industrial, commercial, or residential), and the methods employed (traditional models, machine learning, or deep learning). However, despite significant advancements in algorithms and forecasting methods, finite attention has been given to optimizing and modifying the existing architectures used for conducting these studies. Most research has focused on utilizing cloud platforms for data processing and load forecasting. While these approaches offer high computational power, they face challenges such as high latency, data transmission costs, and dependency on centralized infrastructures. Amid these challenges, performing load forecasting at the edge has emerged as an innovative solution. This approach moves data processing and forecasting closer to the data source, reducing latency, enabling real-time analysis, and improving data security. Nevertheless, the number of studies and research efforts in this area is still limited, emphasizing the need for further exploration and the development of lightweight and efficient models capable of operating effectively on edge devices.

The authors in [8] *explores the application of edge computing and federated learning in STLF for smart homes. The researchers address challenges related to privacy preservation and the need for diverse and large datasets for training models by utilizing edge computing and deep learning models, such as LSTM networks. Simulation results conducted using data from 200 homes in Texas, USA, demonstrate that this method improves prediction accuracy while reducing network load by up to 97% in some scenarios. Personalized models trained with local data also outperformed generalized models. This study highlights that combining edge computing with federated learning offers an efficient and secure solution for STLF in smart grids.*

Lekidis and Papageorgioua [9] *proposed a novel edge-based approach for STLF in home energy management systems (HEMS). The proposed method employs the Temporal Fusion Transformer (TFT), a deep learning model specifically designed for time series forecasting. Input data for this model include historical energy consumption from smart meters, environmental conditions (e.g., temperature, humidity, and radiation), and categorical values such as demographic information. The proposed framework is implemented in a multi-access edge computing (MEC) environment, where data are processed locally, eliminating the need to transmit sensitive information to central servers. This architecture consists of three layers: the device layer (data collection), the service layer (data storage and pre-processing), and the application layer (model execution for forecasting). Simulation results conducted on a HEMS system comprising photovoltaic panels, storage batteries, and smart meters demonstrate a forecasting accuracy of 94.1% using the TFT model. Additionally, the proposed edge-based framework reduced latency to 1.3 seconds, a significant improvement compared to central data center implementations.*

Pang et al. [10] *proposed a distributed framework for smart grid load forecasting based on edge computing and matching theory. The main objective of this research is to optimize intelligent power grid systems by reducing system overhead, data transmission delay, and resource consumption. Within this framework, a delay-aware power resource allocation algorithm (RAA) is designed to enhance system efficiency by task partitioning and distributing them across edge devices. Experimental results demonstrate that the proposed algorithm stabilizes system overhead after 40 iterations. Additionally, for fewer than 50 users, the average delay is reduced to less than 13 seconds, outperforming other algorithms. This research significantly contributes to the digitalization and intelligent development of smart grids and serves as an efficient solution for resource allocation and operational cost reduction in future power systems. Savi and Olivadese* [11] *introduces a distributed architecture for STLF based on Edge Computing and Federated Learning. Using LSTM models, energy consumption data is processed locally on edge devices (e.g., smart meters), and local models are aggregated to produce a global model. Federated Learning ensures that sensitive user data remains local, with only model weights exchanged between devices and the central server. Input features for the model include historical energy consumption, calendar information (e.g., day and hour), and weather conditions (e.g., temperature). Simulation results indicate that this approach achieves comparable accuracy to centralized*

architectures while significantly reducing training time and lowering data transmission volume by up to 50 times. In addition to improving scalability and preserving privacy, the architecture enables user clustering based on behavioral or socioeconomic similarities, further enhancing forecasting accuracy. Gao et al. [12] *develop a decentralized federated learning (DFL) framework for load forecasting in residential buildings. In this approach, data is processed locally on edge devices, and load forecasting models are updated by exchanging gradients between smart home agents in each residential unit. They proposes a gradient selection mechanism to improve communication efficiency and prediction accuracy. This mechanism reduces the size and frequency of gradient exchanges between agents while simultaneously enhancing communication performance and forecasting accuracy. Experiments conducted on the Pecan Street dataset demonstrate that the PriResi framework achieves 97% prediction accuracy while preserving user privacy and reducing system runtime by up to 19% compared to existing methods.*

A review of the latest articles related to load forecasting at the edge level reveals a significant gap in fully offline forecasting capabilities at this level. In other words, many existing approaches still require communication with central servers for final data processing, which contradicts the philosophy of edge computing(minimizing dependence on data centers and enabling local processing).

Moreover, some methods require powerful hardware. This problem may restrict the applicability of such models in environments with limited resources or specific constraints, as these settings often require lightweight and adaptable solutions. Therefore, further research is essential to develop load forecasting models capable of operating fully independently and efficiently at the edge level.

3. *DATA COMPRESSION*

Over the past decades, with the advancements in industrial revolutions, the evolution from traditional grids to smart grids has progressed at a remarkable pace. In smart grids, a massive volume of data is collected, analyzed, processed, and transmitted through various devices such as smart meters, RTUs, PMUs, and other equipment. This data include information such as voltage, current, frequency, active and reactive power, and more, sampled and stored at different frequencies. With the expansion of power grids and the increasing number of smart meters, the volume of data generated has grown exponentially. It is estimated that a typical smart meter in the United States generates more than 1 GB of data every day. By considering the deployment of millions of meters, the daily data volume becomes enormous. This vast amount of data creates challenges such as high storage costs, limited bandwidth, and delays in data transmission.

To address these challenges, various solutions have been proposed, among which data compression stands out as one of the most practical approaches. Data compression is the process of reducing data volume by eliminating redundant information or representing it more efficiently, thereby minimizing storage requirements and optimizing data transmission. In general, data compression methods can be classified into two main categories: lossless and lossy. Lossless methods preserve all the original data information and are more suitable for applications requiring highly accurate data, such as power quality analysis. These methods are typically used in scenarios where even minor changes or losses in data can negatively impact analysis or system performance. For instance, in power quality monitoring systems, precise information about harmonic and voltage fluctuations is essential for identifying network issues. Common algorithms for lossless compression include Huffman Coding, Run-Length Encoding (RLE), LZW (Lempel-Ziv-Welch) Compression, and Arithmetic Coding. These algorithms compress data using repetitive patterns without altering the original information. In contrast, lossy methods reduce data volume by eliminating non-essential details. These methods are commonly used in applications where reducing data size is more critical than retaining fine details. Lossy methods can significantly reduce data size; for example, advanced algorithms like Wavelet-Based Compression (DWT) or Frequency Selective Autoencoders have reported data reduction rates of up to 90%. However, employing these methods requires a careful balance between data reconstruction quality and the compression ratio. Common algorithms for lossy compression include Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), Autoencoders, Fourier Transform, and its variants like Fast Fourier Transform (FFT). Figure 1 illustrates the categorization of data compression methods.

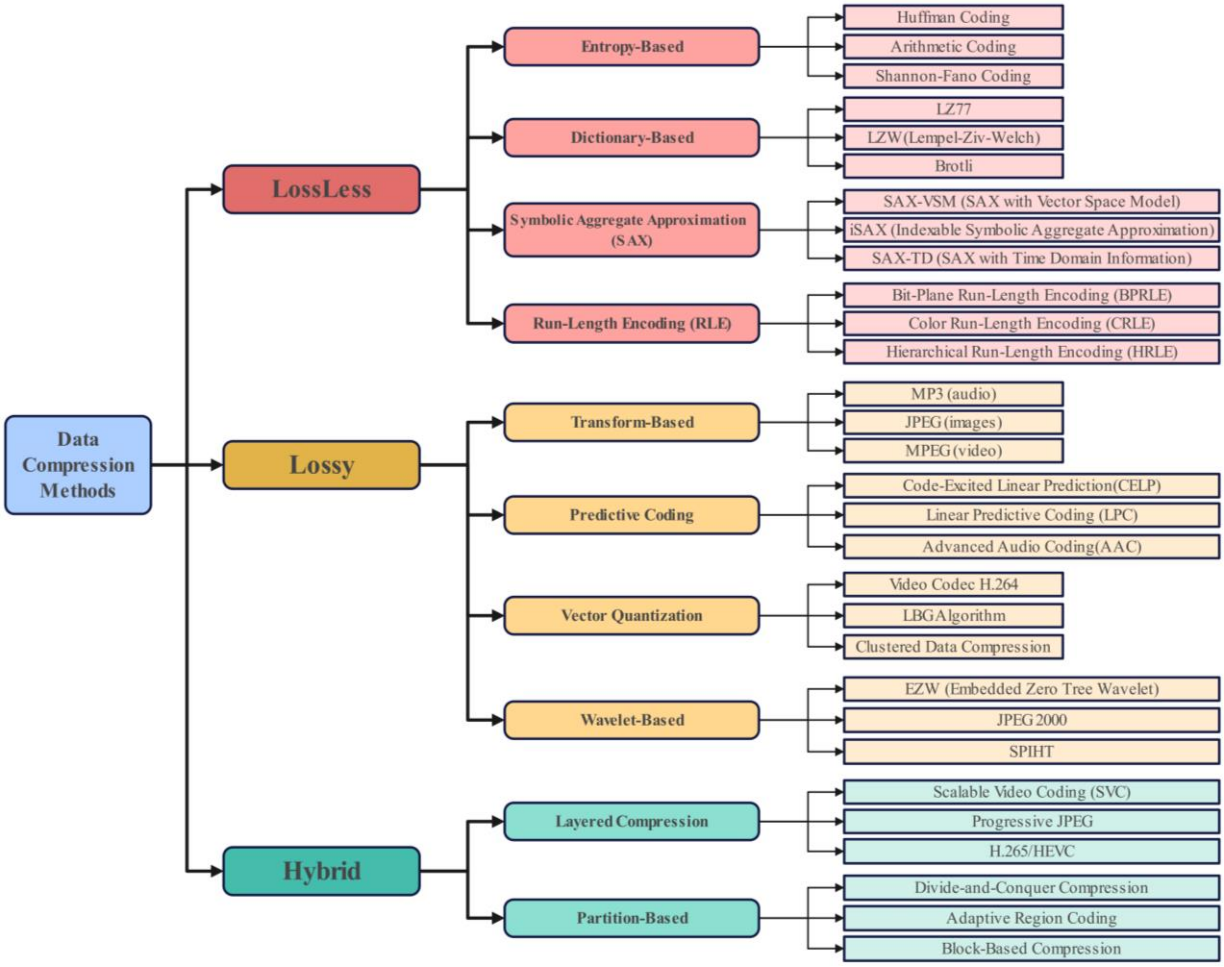

Fig. 1. *Data compression methods*

Like forecasting studies, data compression also employs specific metrics to evaluate the performance of methods. These metrics serve as tools to measure efficiency, accuracy, and the alignment of methods with various application requirements. In data compression methods, evaluation metrics are essential for assessing the effectiveness, quality, and adaptability of these approaches to system needs.

These metrics not only assist developers in identifying the strengths and weaknesses of proposed methods but also provide clear criteria for comparing different algorithms. For instance, in a smart meter system that generates a massive volume of data, compression methods must ensure data volume reduction, preservation of reconstruction accuracy and processing speed. Using appropriate metrics for evaluation can aid in the optimal design of algorithms and find application in various fields such as real-time systems, IoT, and smart grids. Table 1 presents the key metrics for evaluating data compression methods.

Table 1. Data compression metrics

Metric	Evaluation Goal	Formula	Unit	Advantage	Common Applications	Implementation Complexity
Compression Ratio (CR)	Measures data reduction relative to the original size	$CR = \frac{Size_{Original}}{Size_{compressed}}$	Dimensionless	Reduces storage and transmission costs	Data storage and transmission	Low
Compression Gain (CG)	Quantifies the efficiency of compression	$CG = \left(1 - \frac{Size_{Original}}{Size_{compressed}}\right) \times 100$	Percentage	Evaluates compression effectiveness	Optimization of storage and bandwidth	Low
Peak Signal-to-Noise Ratio (PSNR)	Assesses the similarity of reconstructed data to the original	$PSNR = 10\ log_{10} \frac{MAX^2}{MSE}$, MAX: Maximum possible value of the original data	Percentage or dB	Preserves data quality	Critical data analysis	Medium
Mean Squared Error (MSE)	Measures the average squared error between original and reconstructed data	$MSE = \frac{1}{N}\sum_{i=1}^{N}(Data_{Original} - Data_{Reconstructed})^2$	Data unit	Ensures high precision in analysis	Applications like power quality analysis	High
Storage Efficiency (SE)	Calculates the percentage of storage space saved	$SE = \left(\frac{Size_{Original} - Size_{compressed}}{Size_{Original}}\right) \times 100$	Percentage	Reduces storage space for large datasets	Reduces storage space for large datasets	Low
Processing Time	Measures the time required for compression	-	Seconds or milliseconds	Suitable for real-time applications	Real-time systems	High
Energy Consumption	Evaluates energy used during compression	$EC = P \times T(Power \times Time)$	Joules or Watts	Efficient for IoT systems	IoT and low-power devices	Low
Computational Complexity	Quantifies resources needed for processing	-	Operations per unit time	Ideal for low-power systems	Embedded systems	Medium
Throughput	Assesses the amount of data processed per second	$TP = \frac{S_{processed}}{T}$	Bytes/second	Evaluates efficiency in data transmission	Real-time data transfer	Medium

In recent years, with the rapid growth of smart grid technologies and the increasing volume of data generated by different devices, numerous studies have been conducted on data compression to optimize storage, transmission, and processing. These studies aim to reduce costs, enhance efficiency, and maintain data quality. Scalable infrastructures are crucial for managing big data in smart grids, as they enable real-time storage and processing of massive data volumes with minimal cost. These infrastructures enhance system efficiency and ensure network security and stability by leveraging techniques such as data compression and distributed computing [13].

Liu et al. [14] *present a novel framework based on the Divide-and-Conquer approach for compressing and reconstructing smart meter data. The goal is to reduce data volume while preserving accuracy and minimizing processing time. In this method, electrical load data is divided into three categories: event, fluctuation, and steady state, with each category processed using a different compression technique. Event data, due to its critical importance for detecting events, is stored without compression. For fluctuation data, a Compressed Sensing (CS)-based approach utilizing sparse representation and adaptive measurements is employed. Steady-state data is compressed using an improved Symbolic Aggregate Approximation (SAX) method, optimized by the DIRECT algorithm. Experimental results show that this method achieves an average compression ratio of 32.3 and a reconstruction accuracy of 99.7%. Additionally, the processing time for second-interval data is approximately 11 seconds per day, making it suitable for real-time applications.*

Syamsudin et al. [15] *introduce an efficient framework for the compression and classification of power quality disturbances (PQDs) in distributed power systems. The authors utilize three main compression algorithms: Wavelet Transform, Autoencoder, and Convolutional Neural Network (CNN) to process one-dimensional data. Synthetic data for 14 different types of PQDs, generated according to the IEEE-1159 standard, were tested within this framework. The classification process integrates compressed data with the CNN algorithm, achieving high accuracy in identifying PQDs even in noisy environments. The study*

demonstrates that employing CNN for data compression and classification significantly reduces training time and achieves a high accuracy of 99.74%. In contrast, Wavelet Transform and Autoencoder achieve accuracies of 99.52% and 99.03%, respectively, with longer processing times. This framework not only delivers high-speed and accurate processing of PQD data but also highlights that the combination of compression and classification can significantly enhance the performance of deep learning networks.

The authors in [16], propose a data compression algorithm based on improved wavelet transform for power system data. Leveraging the time-frequency localization properties and multi-resolution analysis capabilities of wavelet transform, the proposed method effectively compresses power domain data and enables precise signal reconstruction. The algorithm operates in three main stages: splitting, prediction, and updating, which enhance reconstruction accuracy and minimize reconstruction errors. Experimental results demonstrate that the proposed algorithm outperforms conventional methods and earlier wavelet transform versions. It achieves a compression ratio of 13.52% and a mean squared error of 0.243. Additionally, the energy recovery coefficient of 99.9992% highlights its superior performance. With high processing speed and real-time execution capabilities, the proposed method offers an efficient solution for improving the efficiency of smart power systems while reducing storage and transmission costs.

Wu et al. [17] *introduce a dynamic and parallel two-stage lossless data compression method for smart grid data. The first stage improves the traditional LZW algorithm by incorporating parallel search and dynamic variable-length coding. The second stage combines the improved LZW algorithm with the Huffman algorithm to form a two-stage compression method. The proposed approach was tested on real smart grid data in a MATLAB simulation environment. Results demonstrate that the proposed algorithm significantly outperforms traditional LZW, Huffman, and PDLZW algorithms in terms of compression ratio, compression factor, save percentage, and compression gain, with at least 52% improvement. Additionally, the algorithm reduces storage space effectively, making data transmission more efficient and offering broad applicability for smart grid data processing.*

The authors in [18], *proposes a novel method for smart meter data transmission based on compressed sensing. The method utilizes an over-complete dictionary matrix for sparse representation of data and employs the ROMP algorithm for data reconstruction, reducing reconstruction errors. In this approach, current data from household appliances are sparsely represented and compressed using a random Gaussian matrix as the measurement matrix. The compressed data are then transmitted and reconstructed at the data center. Experimental results demonstrate that this method outperforms traditional techniques such as DWT+OMP and DCT+ROMP in terms of reconstruction accuracy and error reduction. For instance, in tests with appliances such as hair dryers, electric vehicles, and water dispensers, the proposed method achieved an average signal-to-noise ratio (SNR) of 64.04 dB and a mean squared error(MSE) of 0.02. This method not only reduces data transmission costs but also lowers the computational complexity of load identification, making it highly practical for smart power systems.*

By reviewing these articles, it can be concluded that the challenges and research gaps can be categorized into the following areas: delay computation, energy consumption for implementing compression algorithms, reliance on centralized architectures for computations, and the computational complexity of many proposed algorithms. These issues are particularly significant in applications such as IoT systems and smart grids, which face hardware limitations and require real-time processing. One key challenge is the mismatch between the computational power of edge devices and the complexity of compression algorithms. Advanced algorithms, such as deep learning or complex transforms, though offering high accuracy and compression rates, are often impractical to execute on devices with limited power, memory, and bandwidth. This results in increased energy consumption and longer execution times, which can be critical for systems like smart homes and renewable energy stations. On the other hand, using centralized architectures for performing compression computations, while providing greater computational power, leads to higher delays, increased data transmission costs, and security risks. These problems are particularly relevant in distributed and real-time environments, such as load management in smart grids. Ultimately, the need for developing lightweight and efficient algorithms that align with the computational and energy constraints of edge devices is recognized as a major research gap. These algorithms should be capable of delivering suitable compression rates while minimizing delay and energy consumption and maintaining data reconstruction quality. Furthermore, integrating these algorithms with distributed architectures such as edge computing can offer a sustainable and effective solution to address these challenges.

4. *ENCRYPTION*

As mentioned in previous sections, smart meters have emerged as indispensable components of modern power systems. By continuously measuring and transmitting sensitive data such as energy consumption, usage patterns, and time-specific information, these devices facilitate real-time monitoring and optimization of grid conditions. Such capabilities are pivotal for maintaining grid stability and efficiency in increasingly complex energy systems. Beyond real-time monitoring, the data generated by smart meters serves as a critical input for advanced studies such as load forecasting. These studies enable accurate predictions of energy demand, which are foundational for downstream applications like economic load distribution, unit commitment, and demand-response management.

However, the integration of smart meters into power grids introduces significant challenges, with data security and privacy ranking among the most critical concerns. The sensitive nature of the information collected by smart meters necessitates encryption mechanisms to prevent unauthorized access or misuse. For instance, consumption patterns can reveal personal information about users, such as their daily routines, appliance usage, or even occupancy status. The unauthorized disclosure of such data poses severe privacy risks and undermines user trust in the system. Consequently, encryption has become a cornerstone of data security in smart grids, ensuring the confidentiality, integrity, and authenticity of transmitted and stored information.

The unauthorized disclosure of such information not only compromises individual privacy but also undermines trust in smart grid technologies. Public skepticism about data security can become a barrier to widespread adoption of smart meters, which are essential for the transition to modern, efficient power systems. Therefore, addressing these challenges is imperative to ensure user confidence and protect against potential misuse of data. To mitigate these risks, encryption mechanisms have become a cornerstone of data security in smart grids. Encryption ensures that data transmitted between smart meters, substations, and central systems remains confidential, preventing eavesdropping and tampering. Advanced encryption protocols, such as end-to-end encryption, are designed to secure both data in transit and at rest, ensuring that even if data is intercepted, it remains unintelligible to unauthorized parties. Additionally, techniques like digital signatures and secure key management systems are employed to verify the authenticity and integrity of the data, preventing alterations or spoofing attacks. Beyond encryption, other measures such as data anonymization and aggregation are being explored to enhance privacy. Anonymization techniques remove or mask identifiable information, while data aggregation combines data from multiple users to obscure individual patterns. These methods not only reduce privacy risks but also allow utility companies to analyze trends and make informed decisions without compromising individual user identities.

The encryption process begins with key generation and management, where unique cryptographic keys are securely generated for devices and servers. These keys play a vital role in both encrypting and decrypting data. Secure key distribution protocols are implemented to mitigate the risk of interception during communication. Following key generation, sensitive data is encrypted using cryptographic algorithms. Symmetric encryption algorithms like AES (Advanced Encryption Standard) are favored for their speed and efficiency in encrypting large volumes of data. Alternatively, asymmetric algorithms like RSA (Rivest–Shamir–Adleman) are used in scenarios requiring secure key exchange. Once encrypted, the data is transmitted over secure communication channels such as TLS (Transport Layer Security) or IPSec (Internet Protocol Security). These protocols ensure that even if data packets are intercepted during transmission, their contents remain inaccessible to unauthorized entities. At the destination, the data is decrypted using the corresponding keys, restoring it to its original, usable form. This bidirectional process ensures that only authorized parties can access or interpret the data.

Table 2 categorizes practical cryptographic methods based on their type, key length, security level, processing speed, energy consumption, applications, and specific advantages. This classification facilitates the optimal selection of cryptographic algorithms for various applications, particularly in smart grids and smart meters.

Table 2. Cryptographic Algorithm Comparison

Algorithm	***Type***	***Key Length (Bits)***	***Security Level***	***Processing Speed***	***Energy Consumption***	***Applications***	***Advantages***
Advanced Encryption Standard (AES)	*Symmetric*	128/192/256	*High*	*Very Fast*	*Low*	*IoT devices, Real-time data encryption*	*Widely adopted, efficient, and secure*
Data Encryption Standard (DES)	*Symmetric*	56	*Low (outdated)*	*Fast*	*Moderate*	*Legacy systems, low-security applications*	*Simple design*
Blowfish	*Symmetric*	32 – 448	*Moderate*	*Fast*	*Low*	*Embedded systems, password storage*	*Adjustable key size*
ChaCha20	*Symmetric*	256	*Very High*	*Very Fast*	*Very Low*	*Mobile devices, secure communications*	*Resistant to timing attacks*
Twofish	*Symmetric*	128/192/256	*High*	*Moderate*	*Low*	*Embedded systems, file encryption*	*Open-source and flexible*
Triple DES (3DES)	*Symmetric*	168	*Moderate*	*Slow*	*High*	*Financial services, secure communications*	*Enhanced security over DES*
RC4 (Rivest Cipher 4)	*Symmetric*	*Variable* (40 – 2048)	*Low (deprecated)*	*Very Fast*	*Moderate*	*Streaming encryption*	*Lightweight and fast for low-security tasks*
Caesar Cipher	*Symmetric*	*Small (Key = Shift Value)*	*Very Low*	*Very Fast*	*Very Low*	*Educational, low-security applications*	*Simple and easy to implement*
Rivest - Shamir - Adleman (RSA)	*Asymmetric*	1024/2048/4096	*High*	*Moderate*	*High*	*Secure key exchange, digital signatures*	*Strong for key exchange*
Elliptic Curve Cryptography (ECC)	*Asymmetric*	160/256	*Very High*	*Fast*	*Low*	*smart meters Mobile Applications*	*Small key size with strong security*
Diffie-Hellman	*Asymmetric*	2048/4096	*High*	*Moderate*	*High*	*Secure key exchange*	*Simplifies secure key sharing*
Digital Signature Algorithm (DSA)	*Asymmetric*	1024/2048/3072	*High*	*Moderate*	*Moderate*	*Digital signatures only*	*Compact signatures, efficient validation*
Lattice-Based Cryptography	*Asymmetric*	256/512	*Very High*	*High*	*Moderate*	*Quantum-resistant encryption*	*Secure against quantum attacks*

In addition to securing data in transit, encryption plays a critical role in protecting stored data. Whether the data is archived on local devices or centralized servers, storage encryption mechanisms safeguard it against unauthorized access or tampering. For instance, advanced techniques like database encryption and hardware security modules (HSMs) are commonly employed to secure data at rest. By implementing these comprehensive encryption strategies, smart grids can ensure the secure and efficient operation of power grids. As the energy sector becomes increasingly digitalized, encryption remains a fundamental pillar of resilience, enabling the transition to a more sustainable and secure energy ecosystem. The authors in [19] *addresses the challenges of security and privacy preservation in IoT-enabled smart metering systems. The authors introduce data aggregation methods to reduce network traffic while safeguarding user privacy. Fully Homomorphic Encryption (FHE) and Secure MPC are employed to process encrypted data, enabling mathematical operations on the encrypted information. Both approaches face challenges such as message complexity and*

data size. The paper proposes novel protocols for adapting these technologies in AMI. These protocols encrypt measurement data from smart meters and aggregate it hierarchically without revealing the actual values. Simulation results show that the Secure MPC-based protocol is a reliable option for privacy-preserving data aggregation, offering comparable performance to the Paillier encryption system and greater efficiency than FHE. This research represents a significant step toward enhancing the security and efficiency of smart metering networks. Hseiki et al. [20] *focuses on addressing the growing cybersecurity challenges in smart energy meters (SEMs), which are critical components of smart grid. It proposes a secure and resilient SEM design that enhances data integrity, prevents cyberattacks such as Distributed Denial of Service (DDoS), and protects against energy theft. The study explores the evolution of SEMs from conventional meters to modern designs, highlighting their functionalities, vulnerabilities, and comparative advantages. The proposed SEM incorporates LoRaWAN technology for secure and efficient communication while employing a dual-computing unit architecture to isolate critical data processing from user-facing operations. This design ensures robust security by integrating tamper detection, unidirectional data transfer, and real-time monitoring capabilities. Practical implementation results demonstrate the SEM's effectiveness in maintaining data integrity and mitigating cyber threats, making it a comprehensive solution for the evolving needs of smart grids. Du et al.* [21] *introduces a novel authentication method designed for smart meters in the context of smart grids. It focuses on addressing the dual challenges of securing data transmission and reducing computational overhead. The proposed scheme utilizes the Chinese Remainder Theorem (CRT) to enhance the efficiency of identity authentication, enabling lightweight operations suited for resource-constrained environments. Key contributions include the ability to revoke individual smart meter access by leveraging random secret values embedded in hash functions, alongside employing Elliptic Curve Cryptography (ECC) for data encryption. The scheme also addresses various security threats, such as passive attacks, replay attacks, and identity spoofing, by combining hash functions, timestamps, and modular arithmetic to secure data integrity and confidentiality. Performance evaluations reveal the scheme's superior computational and communication efficiency compared to existing methods, making it well-suited for large-scale smart grid deployments.*

Based on these studies, the necessity of encryption can be categorized into four main areas: preserving user privacy, preventing cyberattacks, ensuring data integrity, and protecting critical infrastructure. Additionally, the most significant challenges of encryption in smart meters include limited hardware resources, processing time constraints, balancing security and performance, and key management.

5. *LOAD FORECASTING*

Load forecasting is a critical aspect of modern energy management systems, enabling grid operators to predict future energy demand accurately. This process ensures the efficient allocation of resources, optimal scheduling of generation units, and reliable operation of power systems. By anticipating fluctuations in energy consumption, load forecasting minimizes operational costs, prevents overloading, and enhances grid stability. It also serves as a foundational tool for other applications, such as economic load dispatch, unit commitment, and demand-response programs. With the increasing integration of renewable energy sources and the complexities introduced by distributed energy resources, the accuracy and adaptability of load forecasting methods have become more vital than ever.

STLF specifically focuses on predicting energy demand for time horizons ranging from minutes to several hours ahead. This type of forecasting is particularly crucial for real-time grid operations, as it aids in immediate decision-making for load balancing and resource allocation. STLF relies on analyzing real-time and historical data, including weather conditions, time of day, and consumption patterns, to identify demand trends. In decentralized energy management systems, where data from smart meters and edge devices are used, STLF enables localized decision-making and reduces reliance on centralized systems. In Table 3, the forecasting evaluation metrics have been categorized. It is noteworthy that the actual values (y_i) represent the observed data points, while the predicted values ($\hat{y}_i$) are the estimates generated by the forecasting model. These metrics are computed over n observations to evaluate the model's accuracy and performance.

Table 3. Forecasting accuracy metrics

Metric	Purpose	Formula	Interpretation	Range	Advantages	Disadvantages
Mean Absolute Error (MAE)	measures the average magnitude of errors in a set of predictions.	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$	Lower MAE values reflect higher prediction accuracy	[0,∞)	Easy to understand and compute.	Lower values indicate higher forecasting accuracy.
Mean Squared Error (MSE)	Penalizes larger errors more than MAE to emphasize significant deviations.	$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	Lower values indicate higher accuracy. (0 indicates perfect predictions)	[0,∞)	Highlights significant deviations effectively	Sensitive to outliers
Root Mean Squared Error (RMSE)	Represents the standard deviation of prediction errors.	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	Lower values indicate better model performance.	[0,∞)	Provides errors in the same units as the target variable	Amplifies the impact of outliers
Mean Absolute Percentage Error (MAPE)	prediction error as a percentage of the actual values	$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert \times 100$	Expresses errors as a percentage of actual values.	[0,∞)%	Easy to interpret	Sensitive to small actual values
Symmetric Mean Absolute Percentage Error (sMAPE)	addresses the limitations of MAPE by ensuring a symmetric calculation that treats over- and under-predictions equally.	$\frac{1}{n}\sum_{i=1}^{n}\frac{\lvert y_i - \hat{y}_i\rvert}{(\lvert y_i\rvert + \lvert\hat{y}_i\rvert)/2} \times 100$	Low sMAPE values(close to 0%) indicate accurate predictions with minimal error.	[0,100]%	Less sensitive to extreme outliers than MAPE.	Sensitive to values near zero, can lead to large percentage errors.
Coefficient of Determination (R^2)	Explains the proportion of variance captured by the model.	$R^2 = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$	Values close to 1 indicate a better fit. Negative values show poor fit.	[0,1] or (−∞,1]	Indicates overall model performance	Does not indicate the magnitude of prediction errors
Mean Bias Error (MBE)	Measures average bias in predictions (under or overestimation).	$MAE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$	Values close to 0 indicate unbiased predictions.	(−∞,∞)	Highlights systematic bias in predictions	Cannot measure the size of errors
Forecast Skill Score (FSS)	Evaluates relative improvement over a benchmark model.	$FSS = 1 - \frac{MSE_{model}}{MSE_{benchmark}}$	Values closer to 1 indicate higher model skill.	(−∞,1]	Enables comparison to benchmark models.	Requires a well-defined benchmark

6. *PROPOSED SCHEME*

In the previous sections, the role and importance of data compression and encryption in enhancing the performance of existing systems in smart grids were discussed. Data compression was emphasized as a solution for reducing the massive volume of data generated by smart meters, while encryption was highlighted as a critical measure to ensure data security and privacy. Additionally, the significance of load forecasting, particularly STLF, was addressed as a vital requirement for optimizing energy resource management and preventing sudden network overloading.

This section introduces the proposed framework aimed at addressing two major challenges in smart grids. The framework is structured into three main phases. The first phase focuses on the design and implementation of the proposed smart meter hardware, specifically tailored to support efficient data compression and encryption processes. The second phase elaborates on the computational algorithms employed in the smart meter, incorporating advanced techniques for real-time and optimized data compression and encryption. Finally, the third phase evaluates the results obtained from implementing the proposed framework in either a simulated or real-world environment, analyzing key metrics such as prediction accuracy, compression ratio, transmission delay, and energy consumption.

This paper presents a novel framework to address two critical challenges in smart grids: reducing the substantial volume of data generated by smart meters through efficient compression algorithms and enhancing data transmission security using advanced edge-level encryption techniques. The proposed framework not only improves the system's security and efficiency but also provides a sustainable and reliable solution for data management and ensures the stable operation of the smart grid.

6.1 Hardware

The proposed system is built on a two-layer PCB with dimensions of 5" × 4.2", designed to integrate the functionalities required for advanced smart meter operations. The hardware of the proposed smart meter is illustrated in Figure 2.

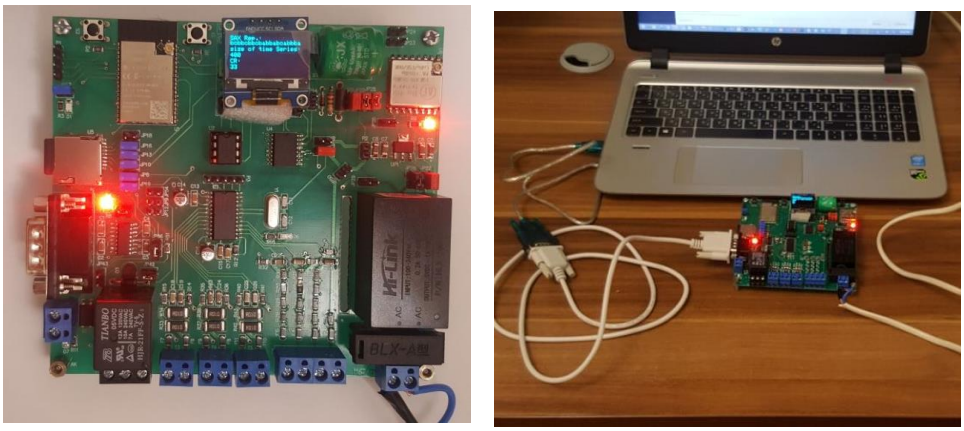

Fig. 2. smart meter hardware

The system's core is an ESP32-WROVER-E, a powerful System-on-Chip (SoC) featuring a 32-bit dual-core CPU, 8MB SRAM, 4MB Flash memory, and built-in connectivity for Wi-Fi and Bluetooth, alongside interfaces like I2C, SPI, and UART. The ESP32 supports adjustable clock frequencies ranging from 80 MHz to 240 MHz and includes a low-power co-processor, making it ideal for IoT platforms, robotics, and wearable devices. For energy metering and monitoring, the system incorporates the ADE7758, a highly specialized IC designed for precise energy metering and power monitoring. The ADE7758 can measure parameters such as active and reactive power, voltage, and current across a dynamic range of 1000:1, ensuring accurate data collection under varying load conditions. To ensure accurate time synchronization, the system includes the DS3231, a highly accurate real-time clock (RTC) IC. The Clock section is also equipped with a backup battery to maintain uninterrupted time tracking during power outages. Its versatility extends to applications like data logging, industrial automation, and scheduling, making it a critical component in the overall design.

The system also incorporates a display unit, an OLED screen, which offers excellent visual clarity even in dim lighting conditions. With low power consumption and a wide viewing angle, it is perfectly suited for applications where energy efficiency and readability are priorities. For communication, the system employs a LoRa module, leveraging long-range, low-power wireless technology to transmit and receive data reliably. Operating on various frequency bands, such as 433 MHz or 868 MHz (depending on regional regulations), the LoRa module ensures efficient data exchange across extended distances, making it ideal for remote monitoring applications. The hardware includes a robust power supply unit (PSU) capable of converting grid voltage to a stable 5V DC, ensuring consistent operation of all system components. Additional circuits for current and voltage measurement, facilitated by the ADE7758, further enhance the system's ability to perform real-time monitoring and data analysis with high accuracy. By focusing on data accuracy, security, and communication efficiency, it provides a comprehensive solution for modern smart metering, ensuring scalability, reliability, and energy conservation for smart grid applications.

6.2 Proposed Algorithm

In the proposed framework, energy consumption data are first processed using time series decomposition methods for load forecasting, enabling the identification of consumption trends and patterns with high precision. Following the forecasting stage, the data undergo compression using two methods (RLE and Huffman Coding) to reduce data volume and optimize storage and transmission efficiency. Subsequently, for ensuring secure transmission, the AES algorithm is employed to encrypt the data, safeguarding it against unauthorized access. Once encrypted, the data are stored locally on an SD card integrated into the smart meter. This additional layer of reliability ensures that even in cases of transmission errors or server failures, the data remain accessible and can be retrieved with precision. This approach not only enhances the fault tolerance of the system but also reduces dependency on continuous server connectivity. The stored data can then be transmitted to central servers as needed, based on environmental conditions or operational demands, ensuring seamless integration with broader energy management frameworks. This multi-layered process effectively balances accuracy, efficiency, and security, paving the way for a robust and decentralized energy management system. Each stage of the process is examined in detail in the subsequent sections, covering methodologies, implementation challenges, and performance evaluations.

6.2.1 Load forecasting

In the proposed framework of this study, the load forecasting phase is designed to execute all computations directly at the edge, specifically within the smart meter itself. This edge-level processing reduces reliance on centralized systems, ensuring faster, localized predictions with minimal latency. To achieve this, a Moving Average (MA) method has been utilized for load forecasting. The MA approach works by smoothing out fluctuations in energy consumption data over a predefined window size, effectively identifying trends and patterns in the historical data. For example, the smart meter aggregates past consumption values within a specific time frame, calculates their average, and uses this value as the forecast for the next time step. This process is repeated iteratively, updating predictions in near real-time as new data becomes available.

The use of the MA method in this context provides a balance between computational simplicity and forecasting accuracy, making it ideal for resource-constrained edge devices like smart meters. Additionally, the Auto-Regressive (AR) method can be employed as a more sophisticated alternative for scenarios requiring higher prediction accuracy. While the MA method excels in its simplicity and low computational requirements, the AR method captures dynamic consumption patterns more effectively by considering the relationships between past data points and current energy usage. By performing these computations at the edge, the system not only enhances privacy but also reduces data transmission overhead and supports a decentralized energy management framework. Leveraging the computational power and connectivity of the ESP32, both MA and AR methods can be implemented efficiently. The MA method provides a lightweight and energy-efficient solution for stable environments, whereas the AR method offers improved accuracy in dynamic scenarios, making the combination of these approaches a versatile tool for edge-based load forecasting in decentralized systems.

Centralized load forecasting is a traditional approach in power grid management, where all consumer data is transmitted to central servers for processing, and the forecasting results are then distributed back to the network. This method, due to access to powerful processing resources and large volumes of aggregated data, typically offers high prediction accuracy. Moreover, the ability to combine data from multiple sources enables deeper analysis and optimized outcomes. However, this approach faces challenges such as delays in data transmission, high communication infrastructure costs, and the risk of privacy breaches. Additionally, its heavy reliance on centralized infrastructure can jeopardize system stability during critical situations like communication outages. In contrast, edge-level load forecasting utilizes edge computing technologies to process data locally, at the source of generation (e.g., smart meters). This approach minimizes the need to transmit data to central servers, thereby reducing delays in analysis and prediction while enhancing data security through localized processing. It also lowers communication costs and enables personalized services for consumers. However, limited computational and energy resources on edge devices may affect the accuracy and performance of forecasting models. Despite these limitations, this method is a scalable and efficient solution for distributed smart grids and decentralized energy management systems.

6.2.2 Data Compression

As previously mentioned, there are various methods available for data compression. Considering this, the focus of this study is on implementing the entire computational process at the edge level. This approach significantly reduces reliance on centralized systems and enhances data privacy. Given the limited computational resources typically available on edge devices, such as smart meters, the selection of algorithms that require minimal computational power while maintaining high processing speeds becomes crucial. Therefore, lightweight and optimized algorithms are essential to ensure smooth operation and scalability. Two practical methods for data compression that not only impose minimal computational overhead but also deliver satisfactory performance are Run-Length Encoding(RLE) and Huffman Coding. These methods are particularly popular for resource-constrained edge devices, such as smart meters, due to their simplicity and low computational requirements.

The RLE method compresses data by converting sequences of repeated values into a single value along with its repetition count, significantly reducing data size for datasets with consistent or repetitive patterns. For example, in energy consumption monitoring, if a smart meter records constant power usage over several time intervals, RLE can compress the data by representing the repeated values with a single entry and the count of occurrences. This approach not only minimizes the storage requirements but also reduces the bandwidth needed for data transmission, making it particularly effective for edge devices with limited resources. Additionally, RLE is inherently robust against noise within the data, as small variations can often be normalized before encoding to maximize compression efficiency. This makes it a practical choice for applications requiring frequent data updates without significantly increasing processing overhead. Figure 3 illustrates the pseudocode for the RLE method.

Pseudocode for Run-Length Encoding (RLE)

Input: Array of data elements, data[ ]

Output: Compressed array of run-length encoded data, encodedData[ ]

Initialize an empty list encodedData[ ]
Set count = 1

For i from 1 to length(data) - 1:
    If data[i] == data[i-1]:
        Increment count
    Else:
        Append (data[i-1], count) to encodedData[ ]
        Set count = 1
Append (data[length(data)-1], count) to encodedData[ ]
Return encodedData[ ]

Fig. 3. Pseudocode of Run- Length Encoding(RLE)

However, while RLE is highly effective for repetitive data, optimizing its performance for edge devices may involve preprocessing techniques, such as filtering out minor fluctuations, to ensure consistent runs of data are better compressed.

The Huffman Coding method is a widely used lossless data compression technique that operates by assigning shorter binary codes to more frequently occurring values and longer codes to less frequent ones. This approach ensures that the total length of the encoded data is minimized, making Huffman Coding highly efficient for datasets with skewed frequency distributions. Unlike fixed-length encoding methods, Huffman Coding uses variable-length codes based on the probability of occurrence, ensuring optimal compression tailored to the specific dataset. The process begins by analyzing the frequency of each value in the dataset and constructing a binary tree, known as the Huffman Tree. Each leaf of the tree represents a value, and its path from the root determines the binary code assigned to it. More frequent values are placed closer to the root, resulting in shorter codes, while less frequent values are assigned longer codes further from the root. This hierarchical structure guarantees that no code is a prefix of another, ensuring proper decoding. Figure 4 illustrates the pseudocode for the Huffman Coding method.

Pseudocode for Huffman Coding

Input: Array of characters and their frequencies, freq[ ]

Output: Huffman Tree and Huffman Codes

Create a priority queue (min-heap) with nodes for each character and its frequency.

While the queue has more than one node:

    Remove the two nodes with the smallest frequencies.

    Create a new node with:

        Frequency = sum of the two smallest nodes.

        Left child = first node, Right child = second node.

    Insert the new node back into the queue.

The remaining node **is** the root of the Huffman Tree.

Generate Huffman Codes:

    Traverse the tree:

        Append '0' **for left** and '1' **for right** traversal.

        Assign codes **to each** leaf node.

Return the Huffman Tree and codes.

Fig. 4. Pseudocode of Huffman Coding

Huffman Coding is particularly effective for datasets with repetitive or unevenly distributed values, such as periodic energy consumption patterns where certain values (e.g., peak or base load levels) occur more frequently. When implemented on resource-constrained hardware like the ESP32, Huffman Coding offers a balance between compression efficiency and computational feasibility. The ESP32's processing capabilities can handle the creation of the Huffman Tree and the encoding process efficiently, provided the dataset size is within the microcontroller's memory limits.

6.2.2. Encryption

Similar to the data compression phase, the encryption phase prioritizes two critical aspects: computational complexity and processing speed. These factors are crucial in ensuring that the encryption algorithms are not only secure but also efficient enough to be implemented on resource-constrained edge devices, such as smart meters. Low computational complexity allows the encryption process to be carried out without significantly burdening the device's processing capabilities, ensuring smooth operation and minimal delays.

The Advanced Encryption Standard (AES) is a robust symmetric encryption algorithm widely recognized for its balance of security, efficiency, and adaptability. Designed to operate on fixed block sizes of 128 bits, AES supports key lengths of 128, 192, or 256 bits, making it highly secure against brute-force attacks. Its lightweight computational requirements and high throughput make it particularly suitable for resource-constrained environments like smart meters and edge devices, where processing power and energy efficiency are critical.

The AES encryption process involves multiple rounds of transformations, with the number of rounds depending on the key length: 10 for 128-bit keys, 12 for 192-bit keys, and 14 for 256-bit keys. Each round consists of four primary operations designed to ensure data confusion and diffusion. The first step, SubBytes, substitutes each byte in the data block with a corresponding value from a substitution box (S-Box), adding non-linearity and resistance to cryptanalysis. The second step, ShiftRows, shifts rows in the block cyclically, introducing data interdependency. The third step, MixColumns, mixes the columns of the block using a linear mathematical operation, spreading the influence of each byte across the block. Finally, the AddRoundKey step XORs the block with a round key derived from the original encryption key, securing the data with key-dependent transformations. Figure 5, demonstrates the pseudocode for the AES encryption process.

Pseudocode for AES Encryption

Input: Plaintext (P), Key (K)

Output: Ciphertext (C)

Key Expansion:

Derive round keys **from** the initial **key** K.

Initial Round:

AddRoundKey: **XOR** the plaintext **with** the initial round **key**.

Main Rounds (Repeat **for** N rounds, based **on key** size):

SubBytes: Substitute **each** byte using an S-Box.

ShiftRows: Shift rows of the state matrix cyclically.

MixColumns: Mix data **in each** column (skipped in the last round).

AddRoundKey: **XOR** the state matrix **with** the current round **key**.

Final Round:

SubBytes

ShiftRows

AddRoundKey

Output the resulting state matrix **as** Ciphertext (C).

Fig. 5. Pseudocode of AES

AES offers several advantages over other encryption algorithms. Its computational efficiency ensures fast encryption and decryption, making it suitable for real-time applications in energy systems. The algorithm's adaptability to different key lengths allows users to choose their level of security based on system requirements. Additionally, AES's widespread adoption as a global standard has led to extensive validation and optimization, ensuring compatibility with modern hardware and software, including edge devices like the ESP32. Compared to asymmetric encryption algorithms like RSA, which are better suited for key exchange, AES excels in encrypting large datasets due to its speed and lower resource consumption.

6.3 Experimental results

The experimental results provide key insights into the effectiveness of load forecasting and data processing techniques across two distinct scenarios. Scenario 1 employs real-time energy consumption data from a smart meter, emphasizing the feasibility of edge-based processing for localized computations. Scenario 2 uses a publicly available dataset, simulating conditions for lightweight, smaller-scale systems. The study evaluates two forecasting methods, Moving Average (MA) and Autoregressive (AR), focusing on metrics such as prediction accuracy, compression efficiency, latency, and energy consumption.

Additionally, RLE and Huffman Coding are assessed for data compression, alongside the encryption overhead introduced by the AES algorithm. These analyses aim to determine the trade-offs between computational simplicity, latency, energy efficiency, and accuracy in a decentralized energy management framework. The results of the forecasting evaluation and the performance comparison of the methods are shown in Figures 6 and 7, respectively.

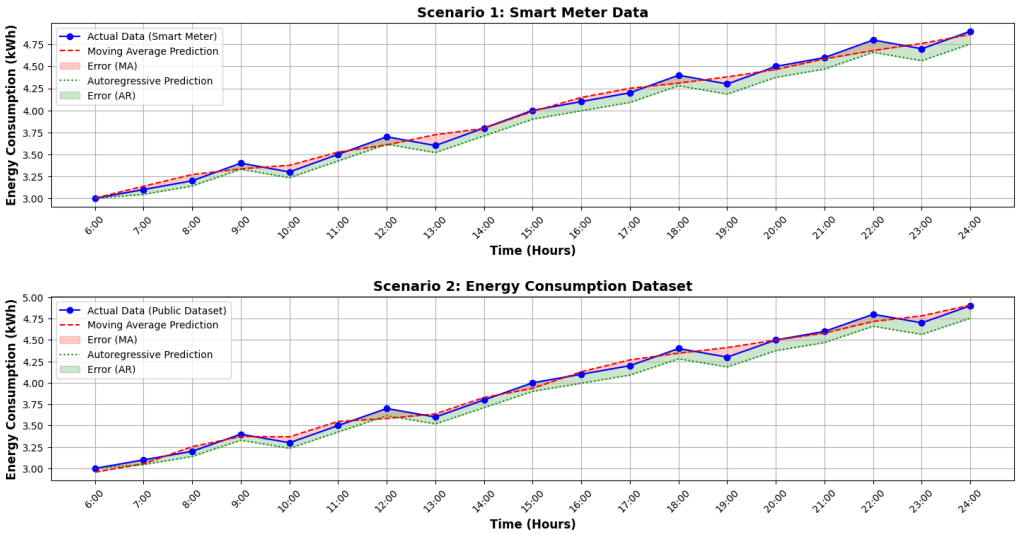

Fig. 6. Evaluation of Forecasting Methods Across Scenarios

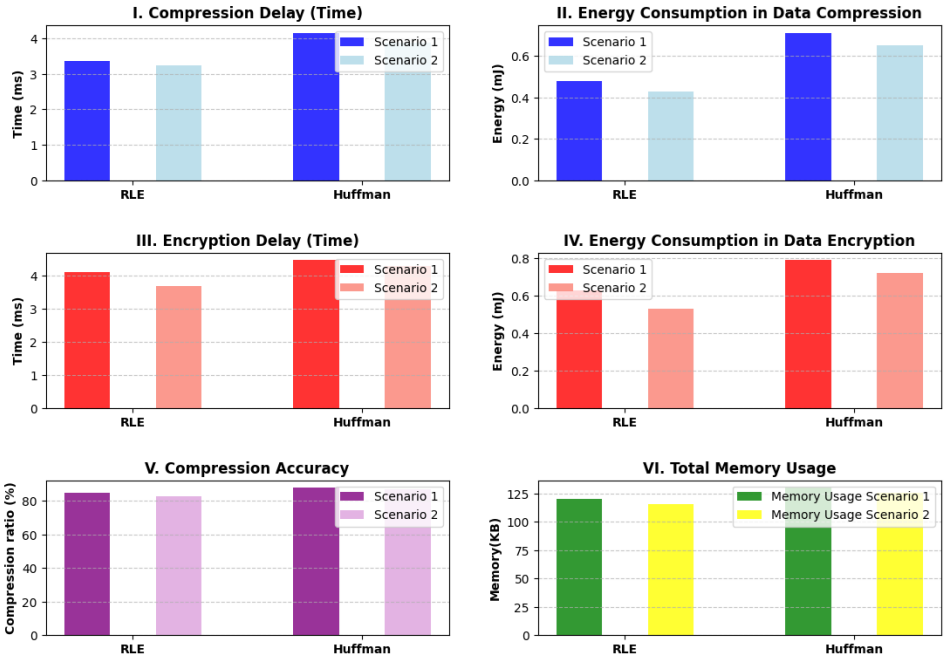

Fig. 7. Comparison of Compression and Encryption Performance Metrics

The forecasting analysis demonstrates that AR model performs better in terms of accuracy compared to the MA method in both scenarios. The AR model consistently achieves lower error margins, with an overall accuracy improvement of around 3-4% compared to the MA method. This highlights the AR model's ability to adapt to dynamic consumption patterns, whereas the MA method struggles with transitional periods, especially at the dataset's start and end. Nevertheless, the MA method is computationally simpler, making it advantageous in environments with stable consumption trends or limited resources. In terms of data compression, Huffman Coding achieves a higher compression ratio compared to RLE(around 5-6% better on average)due to its efficient encoding of frequent patterns. However, RLE compensates with faster processing times and lower energy consumption, with an advantage of approximately 20-25% in these metrics. These findings suggest that while Huffman Coding is ideal for maximum data reduction, RLE is better suited for edge-based systems where

speed and energy efficiency are critical. The encryption phase using the AES algorithm introduces only marginal overhead, ensuring data security without significantly impacting performance. For both compression methods, encryption times differed by less than 10%, and energy consumption remained consistent across both scenarios. This validates AES as a reliable and practical choice for securing edge-based data transmissions. In conclusion, the AR model emerges as the more reliable option for forecasting in dynamic environments, while MA serves as a simpler, resource-efficient alternative for stable datasets. RLE stands out for edge systems requiring quick and energy-efficient compression, whereas Huffman Coding is suitable for scenarios emphasizing maximum data reduction. The AES algorithm effectively secures data, complementing the compression methods. Future improvements could include hybrid forecasting models, adaptive compression algorithms that switch between RLE and Huffman based on data characteristics, and lightweight encryption alternatives to further optimize system performance in decentralized energy management.

CONCLUSION

This study introduces a comprehensive framework that integrates edge-based load forecasting, data compression, encryption, and Multi-Layer Edge Computing (MLEC) to enhance energy management in smart grids. The findings indicate that localized processing through edge computing significantly reduces latency and dependency on centralized systems, enabling real-time decision-making. The Moving Average and Autoregressive models effectively address forecasting needs in stable and dynamic environments, respectively, with the Autoregressive model achieving up to 4% higher accuracy in dynamic scenarios. Data compression methods, such as Run-Length Encoding (RLE) and Huffman Coding, provide complementary advantages in terms of processing speed and data reduction, with Huffman Coding offering approximately 5-6% better compression ratios. Meanwhile, the AES encryption algorithm ensures robust security with minimal computational overhead. Collectively, these advancements facilitate a decentralized, efficient, and secure energy management system. Future research could explore adaptive hybrid models and lightweight algorithms to further enhance performance and scalability in diverse energy management scenarios.

REFERENCES

[1] *K. Fida, U. Abbasi, M. Adnan, S. Iqbal, and S. E. Gasim Mohamed, "A comprehensive survey on load forecasting hybrid models: Navigating the Futuristic demand response patterns through experts and intelligent systems," Sep. 01, 2024, Elsevier B.V. doi: 10.1016/j.rineng.2024.102773.*

[2] *L. Zhang and D. Jánošík, "Enhanced short-term load forecasting with hybrid machine learning models: CatBoost and XGBoost approaches," Expert Syst Appl, vol. 241, May 2024, doi: 10.1016/j.eswa.2023.122686.*

[3] *L. Baur, K. Ditschuneit, M. Schambach, C. Kaymakci, T. Wollmann, and A. Sauer, "Explainability and Interpretability in Electric Load Forecasting Using Machine Learning Techniques – A Review," May 01, 2024, Elsevier B.V. doi: 10.1016/j.egyai.2024.100358.*

[4] *G. F. Fan, Y. Y. Han, J. W. Li, L. L. Peng, Y. H. Yeh, and W. C. Hong, "A hybrid model for deep learning short-term power load forecasting based on feature extraction statistics techniques," Expert Syst Appl, vol. 238, Mar. 2024, doi: 10.1016/j.eswa.2023.122012.*

[5] *Y. Eren and İ. Küçükdemiral, "A comprehensive review on deep learning approaches for short-term load forecasting," Jan. 01, 2024, Elsevier Ltd. doi: 10.1016/j.rser.2023.114031.*

[6] *H. Xu, F. Hu, X. Liang, G. Zhao, and M. Abugunmi, "A framework for electricity load forecasting based on attention mechanism time series depthwise separable convolutional neural network," Energy, vol. 299, Jul. 2024, doi: 10.1016/j.energy.2024.131258.*

[7] *F. P.-A. J. F. C. R. S. W. D. Y. Z. Y. Wenlong Liao, "TimeGPT in Load Forecasting: A Large Time Series Model Perspective," 2024.*

[8] *A. Taik and S. Cherkaoui, "Electrical Load Forecasting Using Edge Computing and Federated Learning," Jan. 2022, doi: 10.1109/ICC40277.2020.9148937.*

[9] *A. Lekidis and E. I. Papageorgiou, "Edge-Based Short-Term Energy Demand Prediction," Energies (Basel), vol. 16, no. 14, Jul. 2023, doi: 10.3390/en16145435.*

[10] *X. Pang, X. Lu, H. Ding, and J. M. Guerrero, "Construction of Smart Grid Load Forecast Model by Edge Computing," Energies (Basel), vol. 15, no. 9, May 2022, doi: 10.3390/en15093028.*

[11] *M. Savi and F. Olivadese, "Short-Term Energy Consumption Forecasting at the Edge: A Federated Learning Approach," IEEE Access, vol. 9, pp. 95949–95969, 2021, doi: 10.1109/ACCESS.2021.3094089.*

[12] *J. Gao, W. Wang, Z. Liu, M. F. R. M. Billah, and B. Campbell, "Decentralized Federated Learning Framework for the Neighborhood: A Case Study on Residential Building Load Forecasting," in SenSys 2021 - Proceedings of the 2021 19th ACM Conference on Embedded Networked Sensor Systems, Association for Computing Machinery, Inc, Nov. 2021, pp. 453–459. doi: 10.1145/3485730.3493450.*

[13] *P. K. Yadav, M. Biswal, and H. Vemuganti, "Smart meter data management challenges," in Smart Metering: Infrastructure, Methodologies, Applications, and Challenges, Elsevier, 2024, pp. 221–256. doi: 10.1016/B978-0-443-15317-4.00002-6.*

[14] *B. Liu, Y. Hou, W. Luan, Z. Liu, S. Chen, and Y. Yu, "A divide-and-conquer method for compression and reconstruction of smart meter data," Appl Energy, vol. 336, Apr. 2023, doi: 10.1016/j.apenergy.2023.120851.*

[15] *M. Syamsudin, C. I. Chen, S. S. Berutu, and Y. C. Chen, "Efficient Framework to Manipulate Data Compression and Classification of Power Quality Disturbances for Distributed Power System," Energies (Basel), vol. 17, no. 6, Mar. 2024, doi: 10.3390/en17061396.*

[16] *Y. Zheng, S. Cui, H. Wang, Q. Cong, D. Ai, and Z. Wang, "Research on Power Data Compression Based on Improved Wavelet Transform Method," in Proceedings - 2023 IEEE International Conference on Energy Internet, ICEI 2023, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 393–396. doi: 10.1109/ICEI60179.2023.00082.*

[17] *Y. Wu, S. Xu, C. Xi, P. Nie, W. Jiang, and S. Hashimoto, "A Dynamic and Parallel Two-Stage Lossless Data Compression Method for Smart Grid," IEEE Access, vol. 11, pp. 143475–143485, 2023, doi: 10.1109/ACCESS.2023.3343436.*

[18] *W. Chen, J. He, G. Cai, and D. Luo, "Smart meter data transmission based on compressed sensing," in Journal of Physics: Conference Series, Institute of Physics, 2023. doi: 10.1088/1742-6596/2477/1/012109.*

[19] *R. Yan, Y. Zheng, N. Yu, and C. Liang, "Multi-Smart Meter Data Encryption Scheme Based on Distributed Differential Privacy," Big Data Mining and Analytics, vol. 7, no. 1, pp. 131–141, Mar. 2024, doi: 10.26599/BDMA.2023.9020008.*

[20] *H. A. Hseiki, A. M. El-Hajj, Y. O. Ajra, F. A. Hija, and A. M. Haidar, "A Secure and Resilient Smart Energy Meter," IEEE Access, vol. 12, pp. 3114–3125, 2024, doi: 10.1109/ACCESS.2023.3349091.*

[21] *J. Du, C. Dai, P. Mao, W. Dong, X. Wang, and Z. Li, "An Efficient Lightweight Authentication Scheme for Smart Meter," Mathematics, vol. 12, no. 8, Apr. 2024, doi: 10.3390/math12081264.*