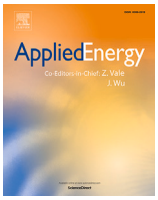
Applied Energy 401 (2025) 126737

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apen



Energy data security and pricing model in local energy markets using artificial intelligence

Fariya Tabassum [a], M. Imran Azim [b,c,*], Md. Rashidul Islam [d,e], M.A. Rahman [f], Liaqat Ali [g], Md. Mahfuzur Rahman [h], M.J. Hossain [e]

[a] Department of Electrical & Computer Engineering, Rajshahi University of Engineering & Technology, Rajshahi 6204, Bangladesh
[b] School of Engineering, RMIT University, Melbourne, VIC, 3001, Australia
[c] Department of Electrical and Computer Systems Engineering, Monash University, Clayton, VIC 3800, Australia
[d] Department of Electrical & Electronic Engineering, Rajshahi University of Engineering & Technology, Rajshahi 6204, Bangladesh
[e] School of Electrical and Data Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia
[f] School of Information Technology, Deakin University, Waurn Ponds, VIC 3216, Australia
[g] Department of Electrical Engineering, Curtin University, Bentley, WA 6102, Australia
[h] Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

HIGHLIGHTS

- AI-powered completely secure framework for IoT enabled smart meters.
- Energy data integrity- and availability-ensured in P2P transactions.
- Along with attack variants, outlier data are considered for robust performance.
- Effective financial benefit allocation among market participants.

ARTICLE INFO

Keywords:
Local energy market
Compromised trading
Secured energy transaction
Customers' profit
Cyberthreat resiliency

ABSTRACT

The increasing adoption of local energy markets has introduced new opportunities for decentralized energy trading but has rendered these systems vulnerable to significant cyberthreats. For local energy markets to remain trustworthy and reliable for efficient energy trading, data availability and integrity must be guaranteed. However, due to the use of contemporary information and communication technologies, these systems are becoming more susceptible to cyberthreats, such as distributed denial of service and false data injection attacks, which can interfere with regular business operations and jeopardize the fairness of trading. This article presents a comprehensive framework utilizing artificial intelligence to ensure a secure bilateral trading environment by identifying corrupted trading data, preventing customers from reacting to it, and mitigating threats' impact on it. In addition, the proposed framework suggests a new real-time optimal trading price-giving model based on artificial intelligence to improve the financial benefits for both sellers and buyers. The framework's effectiveness in maintaining trading data security and operational resilience is demonstrated through a thorough analysis. The simulation results testify that the designed trading price-giving approach benefits both sellers and buyers more than business-as-usual. Moreover, how the secured trading data sharing environment helps in maintaining financial benefits among customers during attack scenarios is also investigated. This work not only enhances the security and dependability of local energy markets but also emphasizes the financial benefits of implementing artificial intelligence-based schemes in energy trading systems.

* Corresponding author at: School of Engineering, RMIT University, Melbourne, VIC, 3001, Australia.
Email address: imran.azim@rmit.edu.au (M.I. Azim).

https://doi.org/10.1016/j.apenergy.2025.126737
Received 14 February 2025; Received in revised form 24 August 2025; Accepted 9 September 2025
Available online 18 September 2025

Nomenclature

$\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_i,d,t}$	SM data shared by prosumer $\mathcal{J}_i \in \mathcal{S}$ at time occurrence $t \in \mathcal{T}$ on a given day $d \in \mathcal{D}$
$\mathcal{Q}^{\mathcal{LC}}_{\mathcal{J}_i,d,t}$	Consumed energy by prosumer $\mathcal{J}_i \in \mathcal{S}$ at time occurrence $t \in \mathcal{T}$ on a given day $d \in \mathcal{D}$
$\mathcal{Q}^{\mathcal{GR}}_{\mathcal{J}_i,d,t}$	Generated energy by prosumer $\mathcal{J}_i \in \mathcal{S}$ at time occurrence $t \in \mathcal{T}$ on a given day $d \in \mathcal{D}$
$\widetilde{\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_i,d,t}}$	Compromised SM information of $\mathcal{J}_i{}^{th}$ prosumer
$\mathcal{O}_t$	Outlier data
$\mathcal{E}_{edi}$	Time elapsed used for FDI breach
$\mathcal{E}_{eddos}$	Time elapsed used for DDoS breach
$\mathcal{A}^{\mathcal{FD}}_{\mathcal{E}_{edi}}$	FDI bias factor
$\mathcal{R}^{\mathcal{E}_{edi}}_{t}$	Random number generated during $\mathcal{E}_{edi}$
$\mathcal{Q}^{\mathcal{E}}_{\mathcal{J}_i\_v}$	Data of victim prosumer
$\mathcal{Q}^{\mathcal{E}}_{\mathcal{J}_i\_f}$	Data of safe prosumer
μ	Mean of the Gaussian distribution
ϱ	Standard deviation of the Gaussian distribution
$\mathcal{M}_{x\_\mathcal{D}_p}$	Maximum depth of DT
$\mathcal{M}_{n\_\mathcal{S}_\ell}$	Minimum sample leaf of DT
$\mathcal{M}_{n\_\mathcal{S}_\mathcal{J}}$	Minimum sample split of DT
σ	Sigmoid activation function
$\mathcal{W}_i$	Input weight of LSTM
$\mathcal{W}_o$	Output weight of LSTM
$\mathcal{W}_f$	Forget weight of LSTM
$\mathcal{W}_c$	Cell state weight of LSTM
$\mathcal{B}_i$	Input bias of LSTM
$\mathcal{B}_f$	Forget bias of LSTM
$\mathcal{B}_o$	Output bias of LSTM
$\mathcal{B}_c$	Cell state bias of LSTM
$\mathcal{k}$	Number of clusters
Θ_i	Centroid of cluster i
$\mathcal{A}_\mathcal{C}$	Accuracy
$\mathcal{P}_\mathcal{C}$	Total number of correct predictions
$\mathcal{P}_\mathcal{T}$	Total number of predictions
$\mathcal{R}_{c\ell}$	Recall
$\mathcal{P}_{on}$	Precision
$\mathcal{S}_{\mathcal{F}1}$	F1 score
$\mathcal{E}_{\mathcal{MA}}$	Mean absolute error
$\mathcal{E}_{\mathcal{RMS}}$	Root mean squared error
$\hat{\mathcal{D}}_{\mathcal{J}oc\_\mathcal{RF}}$	Deviation of the seller's profit, when RF is used, from the original profit during the compromised event
$\hat{\mathcal{D}}_{\mathcal{J}oc\_\mathcal{P}}$	Deviation of the seller's profit, when the proposed model is used, from the original profit during the compromised event
$\hat{\mathcal{D}}_{\hat{\mathcal{b}}oc\_\mathcal{RF}}$	Deviation of the seller's profit, when RF is used, from the original profit during the compromised event
$\hat{\mathcal{D}}_{\hat{\mathcal{b}}oc\_\mathcal{P}}$	Deviation of the seller's profit, when the proposed model is used, from the original profit during the compromised event

Abbreviations

LEMs	Local energy markets
DDoS	Distributed denial of service
FDI	False data injection
AI	Artificial intelligence
BAU	Business-as-usual
RESs	Renewable energy sources
IoT	Internet of Things
SMs	Smart meters
ML	Machine learning
IDS	Intrusion detection system
DL	Deep learning
OCSVM	One class support vector machine
KNN	k-nearest neighbor
BD-LSTM	Bidirectional long short-term memory
DNN	Deep neural networks
CNN	Convolutional neural networks
LSTM	Long short-term memory
RNN	Recurrent neural network
P2P	Peer-to-peer
DT	Decision tree
M-LP	Multi-layer perceptron
k-Mc-NnSA	k-means clustering with nearest neighbor search algorithm
SVM	Support vector machine
RF	Random forest

1. Introduction

The goal of local energy markets (LEMs), which have gained popularity recently as a strategic solution to improve the integration of renewable energy sources (RESs) into the energy grid [1], is to encourage individuals and small businesses acting as consumers, producers or both (prosumers) to engage in energy exchanges within a competitive marketplace [2]. The increasing use of LEMs has revolutionized energy trading by making peer-to-peer (P2P) and decentralized energy exchanges possible [3]. However, the decentralized structure of LEMs renders them vulnerable to various cyber threats [4]. Additionally, the rapid growth of connected devices in Internet of Things (IoT) platforms increases the potential for security vulnerabilities. Therefore, ensuring secure mutual energy transactions is essential [5], particularly in the context of data exchange via IoT devices. The security challenges associated with energy trading [6] are among the technical issues that need to be addressed before implementing a feasible decentralized database-based system, such as second-generation blockchain [7] and third-generation blockchain [8], for bilateral energy trading.

Energy trading amongst residential customers in LEMs is made possible in a significant way by smart meters (SMs) and IoT devices. In order to ensure proper invoicing and identify excess or deficit energy, SMs offer accurate and real-time information on household energy generation and consumption [9]. By combining SMs with domestic appliances and energy management systems, IoT devices improve this ability while facilitating smooth device-to-device connection [10]. This integrated network also optimizes residential energy consumption and automates the monitoring of energy usage patterns. Nevertheless, several security, reliability, and privacy vulnerabilities are brought about by SMs' enhanced networking and communication complexities in LEMs [11]. In particular, the dependability, fairness, and effectiveness of decentralized energy trading in LEMs rely on the security of the SMs' trading data.

Cyber-security issues resulting from vulnerabilities in SMs' communication architecture can be categorized into two key areas—data integrity and availability [12]. Data integrity in the context of energy trading refers to safeguarding trading information from manipulation, corruption, and unauthorized changes. Data integrity breaches, such as those caused by false data injection (FDI) attacks, can corrupt transaction records, resulting in monetary losses and controversies among competitors [13]. Since dynamic energy prices are established using genuine and accurate data that reflect actual supply and demand, maintaining data integrity encourages fair trading and market trust. Data availability, especially in real-time systems like LEMs, guarantees that information is available and usable whenever it is needed. The continuity of energy

trading is at risk due to attacks like distributed denial of service (DDoS), which interfere with data flow and cause transactions to be delayed or fail [14]. Therefore, maintaining the availability and integrity of SMs' trading data is essential in the context of LEMs.

In order to maintain data integrity in SMs, researchers of Ref. [15] have suggested a complex data authentication system that incorporates one class support vector machine (OCSVM) and isolation forest, machine learning (ML) approaches, to identify compromised data. Their designed OCSVM and isolation forest model depict the attack detection accuracy of 91.6 % and 88.33 %, respectively, which should be improved. Moreover, no prevention and attack impact mitigation strategies are discussed. A privacy-preserving protocol for high-frequency smart meters, that uses Paillier homomorphic encryption and difference expansion-based reversible watermarking is employed in [16] to guarantee the integrity of trading data. Their proposed technology is only capable of identifying certain data integrity invasions; it is unable to stop customers from responding to them. In order to detect DDoS attacks targeting energy hubs using IoT devices, a number of supervised machine learning techniques were examined in [17], including decision tree (DT), gradient boosting, support vector machine (SVM), k-nearest neighbors (KNN), and random forest (RF). There is no identification of an integrity attack in this work. Also, there is no strategy for attack impact mitigation or prevention.

In [18], it was suggested that blockchain technology could be integrated with the IoT to address data security from an integrity standpoint. However, there is no discussion on how to prevent data modification beyond the blockchain domain, and the DDoS incident is also overlooked. In [19], a deep learning (DL) model-based intelligent intrusion detection system (IDS) for IoT devices is presented. To enhance the identification and reduction of possible security risks in IoT networks, the performance of deep neural networks (DNN), convolutional neural networks (CNN), long short-term memory (LSTM), and recurrent neural networks (RNN) models is assessed. A CNN and a gated recurrent unit-based hybrid IDS, proposed in [20], is limited to detecting DDoS attacks only. Using the blockchain-based framework, authors have suggested a layered DDoS mitigation strategy to safeguard IoT devices and other computational resources or machines, as explained in [21]. However, this ignores the precautions a user should take after discovering a corrupted event. In [22], authors offered a system based on the classification of anticipated residuals (CPRs) for the detection of FDI attacks by utilizing real-time data captured with edge computing. To improve the detection accuracy, the anticipated residuals are classified after being predicted from the obtained measurement data. Despite increasing the detection rate of FDI attacks, this mechanism does not make recommendations for strategies to prevent or lessen the impact of attacks.

Energy trading's growing scale and complexity in LEM call for sophisticated solutions to guarantee system dependability and cyber-security. Traditional attack detection and mitigation techniques frequently fail in the context of changing attack tactics and real-time trading situations. Although some ML and DL-based methods have also been adopted, their functions are limited to either maintaining data integrity or availability to some extent. This research proposes an artificial intelligence (AI)-based framework that ensures a secure P2P trading platform by identifying compromised trading data, preventing customers from responding to it, and finally retrieving the original trading data. The designed AI-based security approach along with an overview of the different security tactics employed by earlier studies is included in Table 1.

One of the P2P trading's aims is to reduce electricity costs, which may encourage customers to join the LEM framework. To achieve this goal, numerous trading strategies are proposed in the literature. Trading areas based on trading platforms, blockchain, game theory, simulation, optimization and algorithms are discussed by the researchers of Ref. [23–25]. Some researchers focus on optimal trading price suggesting models as it is not only essential for balancing buyer and seller benefits but also critical for encouraging market participation and stability [26]. In [27], a game-theoretic technique is provided for determining the trade price in a P2P-based electricity market. [28] proposes a P2P cooperative trade model based on the generalized Nash bargaining algorithm. A strategy based on game theory and mixed integer linear programming (MILP) optimization is given for market settlement in P2P energy trading in [29]. In order to maintain coalition stability and benefit prosumers, researchers of Ref. [30] decide to use the mid-market rate as the pricing mechanism for P2P trade.

All of these traditional price giving methods suffer from various issues, such as: (a) tactics based on static pricing structures generally fail to adapt to changing market conditions; (b) large P2P energy trading networks with many members make it difficult for rule-based or fixed pricing methods to scale efficiently; (c) many approaches fail to achieve a balance that guarantees equity for all customers, favoring either the buyers' or sellers' interests. Dissatisfaction and decreased market involvement may result from this imbalance; (d) due to DERs and fluctuating consumption patterns, P2P energy markets are naturally unstable. Conventional approaches are unable to handle these uncertainties; and (e) traditional approaches frequently lack strong security features, making them vulnerable to DDoS and FDI attacks. To address these problems, AI can be a useful tool to guarantee dynamic and Adaptive Pricing by analyzing real-time data; financial benefits for both buyers and sellers; optimal solutions for large-scale P2P trading; stable and efficient market outcomes; and cybersecurity.

It is investigated in [31] how distributed machine learning might be used to address the prosumer energy trading issue in a future electric distribution system. [32] proposes a predictive optimization-based nanogrid energy trading model that incorporates particle swarm optimization (PSO) and bidirectional long short-term memory (BD-LSTM) to minimize energy trading costs while providing an energy sharing plan amongst peers. Their prediction model gives RMSE values of 1.45, 1.98 and 1.26 for energy load, energy consumption and PV generation respectively, which can further be improved. The researchers in [33] focus on electricity cost reductions from the P2P market through

Table 1
An outline of security measures for various cyberattacks.

Literature	Scheme	Data unavailability	Data confidentiality	Approaches		
				Detection	Prevention	Impact reduction
[15]	ML	✓	×	✓	×	×
[16]	Encryption	✓	×	✓	×	×
[17]	ML	×	✓	✓	×	×
[18]	Blockchain	✓	×	✓	×	×
[19]	DL	×	✓	✓	×	✓
[20]	DL	×	✓	✓	×	×
[21]	Blockchain	×	✓	✓	×	✓
[22]	CPRs	✓	×	✓	×	×
Proposed	AI	✓	✓	✓	✓	✓

In this case, ✓ and × denote respectively the existence and exclusion in the literature.

demand side management tactics employing a multi-agent deep reinforcement learning approach. Training numerous agents with deep networks needs significant processing power and time, which could hamper real-time decision-making and limit scalability, particularly in resource-constrained edge contexts such as SMs. The authors of Ref. [34] suggested a forecasting framework that uses a number of customized machine learning techniques based on the features of the bidding data. Their proposed solution depends on thorough model personalization. In [35], a profit-driven approach is used to schedule virtual power plant resources during periods of high demand in a competitive energy market by employing several ML techniques to forecast the generating pattern. Several EV Load Curve forecasting approaches, including statistical, machine learning, and deep learning techniques, are examined in [36] in order to offer guidance to participants in the wholesale energy market. To accelerate traditional blockchain-based P2P trade, the authors of Ref. [37] proposed a blockchain and machine learning-based system to lower consensus time. Furthermore, machine learning is employed to determine appropriate lending rates for buyers. Although the suggested system effectively reduces consensus time when compared to standard blockchain systems, convergence time could possibly be further lowered by using a purely AI-driven framework. Furthermore, the model focuses solely on determining lending rates for buyers, ignoring the possible impact and benefits for sellers, limiting its usefulness in a properly balanced peer-to-peer energy trading system.

Motivated by the effectiveness of AI in various aspects, this work adopts a new real-time trading price-giving model based on AI. The aim of this research is to develop a comprehensive AI-driven framework that ensures secure P2P energy trading by leveraging intelligent models for real-time attack detection, preventing participants from reacting to compromised data, mitigating attack impacts, and suggesting optimal trading prices. After detecting any compromised SM's energy data affected by FDI and DDoS, it will immediately be separated from the original trading environment and the participants will be notified, which consequently prevents the customers from reacting to corrupted data. Then, this corrupted data will be stored for further analysis and notified to the control entity. After that, the original energy data is recovered to lessen the attack's impact. Lastly, the framework will offer an optimal trading price to facilitate participants financially.

The framework is designed with a focus on real-time threat identification and impact mitigation in SM contexts with limited computing and memory resources. Therefore, the requirement for lightweight models that guarantee both computational efficiency and interpretability in addition to effective performance served as a guide for choosing models for various layers of the framework. Inspired by the effectiveness of DT, a supervised ML model, in classification tasks along with attack identification [38], this work adopts DT for the attack detection purpose. When a corrupted event is identified, the original trading data is recovered using LSTM, a DL-based model. Finally, a novel scheme that combines the architecture of k-means clustering and nearest neighbor search algorithm (k-Mc-NnSA), is used to offer the real-time optimal trading price. Despite being fundamental models compared to state-of-the-art alternatives, DT, LSTM, and k-Mc-NnSA are selected here in order to meet our framework's design goal, i.e., lightweight models having less computational burden, interpretability, and effective performance within constrained environments. The proposed framework is intentionally referred to as "AI-driven" since it integrates intelligent decision-making models throughout system layers, including unsupervised ML model for attack detection and classification, DL model for retrieving compromised data, and heuristic search approach to provide optimal trading price. The framework's reliance on AI to identify hostile activity, lessen its effects, guarantee secured trading environment, and offer optimal trading price is illustrated here by this terminology. The primary contributions of this research, which focus on proposing a comprehensive framework that takes into account the security of SM's data along with the financial aspects of LEMs, are summarized as follows:

- An AI-driven secure environment for SM's data is proposed to enable real-time attack detection, prevent participants from acting on corrupted data, and mitigate the impact of attacks.
- A new AI-based real-time optimal trading price-giving model is designed to facilitate P2P energy trading, ensuring profitability for customers (buyers and sellers).
- The efficiency of the proposed framework in preserving the security of trading data is demonstrated. Its performance is also evaluated by analyzing trading benefits for customers under both corrupted and proposed cyber-secured LEM environment conditions, proving its robustness along with economic advantages.

The remaining sections of this paper are arranged as follows. A summary of the LEM structure utilized for attack implementation is provided in Section 2. Sections 3 and 4 contain the specifics of possible threat models and the suggested structure for protecting and benefiting the LEM's entities, respectively. Section 5 covers the proposed framework's performance evaluation with experimental set-up. Section 6 investigates the effect of the proposed cyber-secured trading environment on LEM's financial aspects. Finally, the paper is concluded with recommendations for future research in Section 7.

2. Energy market testbed for attack implementation

The suggested framework's resilience against cyberthreats was assessed by creating a LEM testbed that replicates a P2P energy trading environment in the event of cyberattack. The testbed assumed a completely decentralized P2P energy market in which participants deal with one another directly without the need for centralized regulatory supervision. Because of its decentralized structure, the system is more susceptible to cyberattacks by nature, which makes it perfect for evaluating how strong the designed security scheme is. Fig. 1 illustrates the cyber vulnerabilities and the way prosumers exchange trading information in this type of P2P market. The LEM testbed includes 30 customers and among them first 15 have rooftop solar PV panels who take part in trading as sellers based on their availability of energy. P2P participants are supposed to have SMs, capable of recording solar PV generation and energy demand for determining their in-house power status. Again, SMs can track P2P traded quantities and pricing. All participants are supposed to have individual accounts on a distributed ledger platform. These accounts are linked to SMs, which capture P2P trading data and store it on the blockchain. After the P2P trading data has been recorded, smart contracts are assumed to be deployed to enable the settlement of

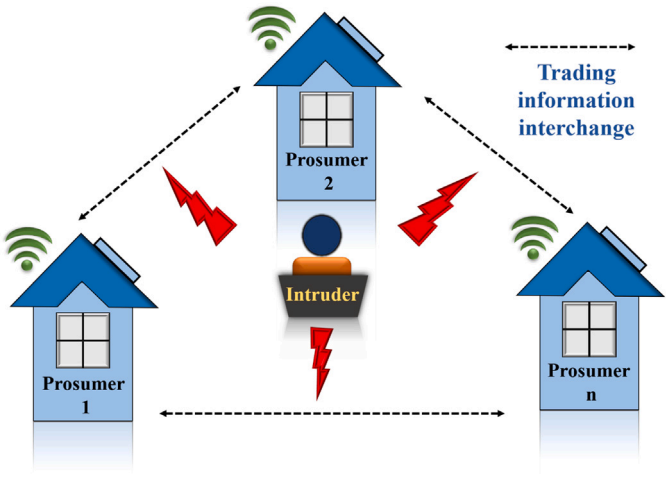


Fig. 1. An schematic of P2P energy trading among entities with possible cyber vulnerability.

buying and selling transactions within the LEM. The energy data available for trading, $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t}$, exchanged by a customer at time occurrence $t \in \mathcal{T}$ on a given day $\mathcal{d} \in \mathcal{D}$, can be expressed as follows in a typical trading system:

$$\mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t} = \mathcal{Q}^{\mathcal{LC}}_{\mathcal{s}_i,\mathcal{d},t} - \mathcal{Q}^{\mathcal{GR}}_{\mathcal{s}_i,\mathcal{d},t} \tag{1}$$

where the energy produced by a customer $\mathcal{s}_i \in \mathcal{S}$ utilizing the accessible RESs is denoted by $\mathcal{Q}^{\mathcal{GR}}_{\mathcal{s}_i,\mathcal{d},t}$, and $\mathcal{Q}^{\mathcal{LC}}_{\mathcal{s}_i,\mathcal{d},t}$ denotes the energy consumed by the customer at a specific moment $t \in \mathcal{T}$. The given criterion of $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t}$ is obtainable from a customer's energy production and consumption:

$$\mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t} < 0, \text{ when } \mathcal{Q}^{\mathcal{LC}}_{\mathcal{s}_i,\mathcal{d},t} < \mathcal{Q}^{\mathcal{GR}}_{\mathcal{s}_i,\mathcal{d},t}, \quad \forall \mathcal{s} \in \mathcal{S}, \forall \mathcal{d} \in \mathcal{D}, \forall t \in \mathcal{T} \tag{2}$$

$$\mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t} > 0, \text{ when } \mathcal{Q}^{\mathcal{LC}}_{\mathcal{s}_i,\mathcal{d},t} > \mathcal{Q}^{\mathcal{GR}}_{\mathcal{s}_i,\mathcal{d},t}, \quad \forall \mathcal{s} \in \mathcal{S}, \forall \mathcal{d} \in \mathcal{D}, \forall t \in \mathcal{T} \tag{3}$$

$$\mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t} = 0, \text{ when } \mathcal{Q}^{\mathcal{LC}}_{\mathcal{s}_i,\mathcal{d},t} = \mathcal{Q}^{\mathcal{GR}}_{\mathcal{s}_i,\mathcal{d},t}, \quad \forall \mathcal{s} \in \mathcal{S}, \forall \mathcal{d} \in \mathcal{D}, \forall t \in \mathcal{T} \tag{4}$$

(2) represents the scenario where a customer $\mathcal{s}_i \in \mathcal{S}$ has excess energy available for trading. In this case, the SM records negative trading data, indicating that the prosumer is acting as a seller, supplying energy to other customers. On the other hand, a metering data value that is positive, denotes lack of energy production, necessitating customer to buy energy from other customers, as expressed in (3). Finally, if the energy produced at any instant $t \in \mathcal{T}$ is enough to fulfill the demand, then the energy transaction represented in (4) occurs. Here, $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t} = 0$, indicating that the participant $\mathcal{s}_i \in \mathcal{S}$ is neither a seller nor a buyer. It is clear from the above market dynamics, (1)-(4), that manipulating SMs data might seriously hamper the trading process, deceive players and jeopardize the efficiency and fairness of the market as a whole.

3. Potential attack template and strategy

Simulating realistic cyberattack scenarios that target a decentralized P2P energy market is crucial to assessing the suggested framework's attack defensive capabilities. This section discusses the two main attack types intended to undermine the market's operation: distributed denial of service (DDoS), which seeks to interfere with data availability, and false data injection (FDI), which focuses on compromising data integrity. An attack deployment strategy is also provided here, detailing the requirements and schedule for carrying out these cyberthreats. By examining these situations, this section highlights the necessity of strong security measures and offers a basis for comprehending how hackers can take advantage of loopholes in IoT-based P2P trading networks and SM systems.

3.1. Violation of data integrity

Since precise and reliable information is necessary to ensure fair trade and market stability, maintaining data integrity is crucial [39]. The integrity of the P2P energy market is threatened by the illegal and clandestine alteration, destruction, or modification of shared SMs data. Data that has been altered or falsified might mislead participants, disrupt the energy balance, and cause losses or inefficiencies in operations. The primary approach used in this work to represent data integrity violations is an FDI attack, in which adversaries alter SM's trade-worthy energy data to deceive market operations and obstruct genuine trade among participants. Through FDI attacks, the information communicated and the decisions required to maintain a reliable system for energy trading can be influenced [40].

FDI commonly simulates the integrity attack using scaling, pulse, random, ramp, and smooth curve [41]. Among these different attack templates, the random attack is used most frequently for implementing FDI attacks to corrupt load or demand data [42]. Furthermore, many real-world cyberattacks are not deterministic, and attackers frequently inject random fake data to make the attack difficult to detect and trace. By incorporating randomization, the attack simulates real-world scenarios in which attackers destroy data without following a specific pattern. Hence, this attack template is employed in this work. In addition, random data corruption techniques are useful in testing the robustness and adaptability of the proposed security framework. A representation of the compromised SM energy data for $\mathcal{s}_i{}^{th}$ prosumer, denoted as $\widetilde{\mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t}}$, under the random attack template pattern based on the discussion in [41] is as follows:

$$\widetilde{\mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t}} = \begin{cases} \mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t} + \mathcal{O}_t, & \forall t \notin \mathcal{E}_{edi} \subset \mathcal{E} \\ \mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t} + [\mathcal{A}^{\mathcal{FD}}_{\mathcal{E}_{edi}} \times \mathcal{R}^{\mathcal{E}_{edi}}_t] + \mathcal{O}_t, & \forall t \in \mathcal{E}_{edi} \subset \mathcal{E} \end{cases} \tag{5}$$

Here, $\mathcal{E}_{edi}$ is the time elapsed used for the entire FDI breach, and $\mathcal{E}$ denotes the total operating time of $\mathcal{s}_i{}^{th}$ customer. If $\mathcal{E}_{\mathcal{s}t} \in \mathcal{E}$ denotes data breaching's start time and $\mathcal{E}_{en} \in \mathcal{E}$ is the end time, then $\mathcal{E}_{edi} = [\mathcal{E}_{\mathcal{s}t}, \ldots, \mathcal{E}_{en}]$. $\mathcal{A}^{\mathcal{FD}}_{\mathcal{E}_{edi}}$ represents the intruder's choice of the bias factor to implement FDI. The outlier data that naturally coexists with the SMs data, $\mathcal{O}_t, \forall t \in \mathcal{E}$, is represented by (5). Section 4, on data preconditioning, contains information concerning this bad data. A possible way to express the random number produced by $\mathcal{R}^{\mathcal{E}_{edi}}_t, \mathcal{E}_{\mathcal{s}t} < t < \mathcal{E}_{en}, \forall t \in \mathcal{E}$, is as follows:

$$\mathcal{R}^{\mathcal{E}_{edi}}_t = \begin{bmatrix} r_{d1} \\ r_{d2} \\ \vdots \\ r_{dn} \end{bmatrix} \tag{6}$$

where $r_{d1}, r_{d2}, \ldots, r_{dn}$ are the elements of $\mathcal{R}^{\mathcal{E}_{edi}}_t$ at $t^{th}, (t+1^{th}), \ldots, (t+n-1^{th})$ moments, respectively, and $\forall t \in \mathcal{E}$. Since SMs serve as the foundation for energy data recording and communication, they need to have strong defenses against this threat.

3.2. Violation of data availability

The timely and dependable access to information that supports the steady and effective functioning of energy markets depends on the availability of trading data. Attacks that target data availability cause delays, disruptions, or even distortions in energy trading operations by interfering with the smooth flow of trading information. The stability, effectiveness, and security of the LEMs' activities may be seriously endangered by such interruptions. DDoS cyberattacks overload the network and prevent the sharing of real-time energy trading data by taking advantage of important nodes or communication routes. By exhausting vital resources like CPU, memory, bandwidth, and server buffer capacity, these attacks attempt to make legitimate services unavailable. DDoS threats, which are frequently planned and target SMs and other IoT devices, have become a serious concern in the context of IoT-enabled LEM platforms [43,44]. It is thought that the intruder makes the targeted customer $\mathcal{s}_i \in \mathcal{S}$ inaccessible for any occurrence $t \in \mathcal{T}$ on a day $\mathcal{d} \in \mathcal{D}$. The victim participant's modified trade-worthy data can therefore be expressed as follows:

$$\widetilde{\mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t}} = \begin{cases} \mathcal{Q}^{\mathcal{TR}}_{\mathcal{s}_i,\mathcal{d},t} + \mathcal{O}_t, & \forall t \notin \mathcal{E}_{eddos} \subset \mathcal{E} \\ \mathcal{O}_t, & \forall t \in \mathcal{E}_{eddos} \subset \mathcal{E} \end{cases} \tag{7}$$

The extensive effects of DDoS on IoT systems and energy trade emphasize how urgently strong remedies are needed to guarantee the reliability and accessibility of energy market activities.

3.3. Attack deployment scheme

To launch a cyberattack on the assumed LEM testbed, as outlined in Section 2, each participant is thought to be using an SM that employs IoT

Table 2
Attack execution plan.

SM data security concern	Attack types	Targeted days of each victim customer for attack						Compromised data by individual attack
		ϑ_5	ϑ_9	ϑ_{11}	ϑ_{13}	ϑ_{22}	ϑ_{26}	
Integrity	FDI	$1^{st}, 2^{nd}$	$2^{nd}, 4^{th}$	$2^{nd}, 3^{rd}$	$4^{th}, 5^{th}$	$5^{th}, 6^{th}$	$1^{st}, 3^{rd}$	4 % of total data
Unavailability	DDoS	$5^{th}, 7^{th}$	$6^{th}, 7^{th}$	$5^{th}, 7^{th}$	$1^{st}, 3^{rd}$	$1^{st}, 2^{nd}$	$4^{th}, 6^{th}$	4 % of total data

technology to collect energy data, including generation and consumption. The attack scenario focuses on manipulating trade-worthy energy data rather than the per-unit energy price, and since such manipulation could seriously disrupt trading, mislead participants, and jeopardize the fairness and efficiency of the market as a whole, these SMs are presumed to be the main target of intruders. This research assumed that there are 30 residential customers in the LEM. With a time horizon of every 5 min, the net metering data of each participant spans a week. There are 288 time slots in the 24-h scheduling period, covering the time starting at 12 a.m. and finishing at 11:55 p.m. A realistic scenario was created in order to carry out cyberthreats, in which adversaries insert FDI and DDoS into a subset of prosumers' SMs. 20 % of the total market participants, 6 in number, are randomly selected to be the attacker's target. While DDoS attempts to interfere with the data transmission network that facilitates P2P trade, FDI manipulates net energy measurements to produce disparities in the published results. Each attack category selects 4 % of a customer's data set for compromise, indicating that when the 2 templates are used, 8 % of each participant's entire data collection is compromised. Table 2 provides the attack strategies for the considered LEM testbed. The 6 victim customers— $\vartheta_5, \vartheta_9, \vartheta_{11}, \vartheta_{13}, \vartheta_{22}$, and ϑ_{26}—shared net metering data are manipulated by the attack templates with varying biasing factors.

FDI allows attackers to falsely overestimate energy generation or suppress consumption statistics. This modified data can then be sold profitably within the LEM, essentially allowing the attacker to obtain monetary benefits without truly producing or conserving energy. Attackers may target competitors by increasing their consumption values, limiting their trade capacity, and giving themselves an unfair market advantage. In contrast, DDoS attacks are driven by the goal of disrupting system operations and weakening the trading platform's trust and reliability. By overwhelming the communication infrastructure or delaying data transmissions, attackers may hinder timely energy trades, generate market instability and prohibit participant involvement. In this work it is considered that, the primary goal of an attacker is to manipulate trading data for personal financial advantage. While targeting any customer for attack deployment purposes, two factors—whether energy is produced through rooftop solar PV panels or not and type of attack (FDI, DDoS)—are considered here. Moreover, it is also assumed that, a positive bias factor is chosen for FDI implementation purposes. However this strategy does not ensure that every attack instance will be profitable for the intruder. A summary of the changing market dynamics along with a financial benefit analysis of attacker from this attack deployment strategy is presented in 5.8.

4. Developed framework for securing energy trading with optimal trading price

This article offers an extensive strategy for securing the two-way energy transactions of decentralized LEM using AI-based techniques. The proposed framework can identify compromised net metering data, prevent customers from reacting to it, and mitigate attack impact while providing the best trading price to increase profitability for both buyers and sellers. Fig. 2 schematically represents the functioning of the proposed framework whereas Fig. 3 shows how the trading data security layer works to identify and prevent corrupted energy data before it enters the trading stage.
When the trading data is benign, attack identification model will detect it as "Real", as shown in Fig.3a, and it will be directly used as trading data. Fig.3b illustrates the situation in which the attack detection model identifies a corrupted value (FDI/ DDoS) in the energy data, the impact mitigation model replaces it with a forecasted value, and the data is then sent as trading data. The following stages describe how the framework functions:

- **Detection**: Initially, an ML-based model called DT, detects abnormalities in participants' shared energy data that are symptomatic of attacks such as DDoS and FDI.
- **Prevention**: To ensure the compromised data cannot affect the choices made by market participants or the functioning of the system, it is filtered into a junk folder as soon as an attack is discovered. By notifying the participant and the central control entity of the data type for additional analysis, this step prevents the customer from reacting to distorted data.

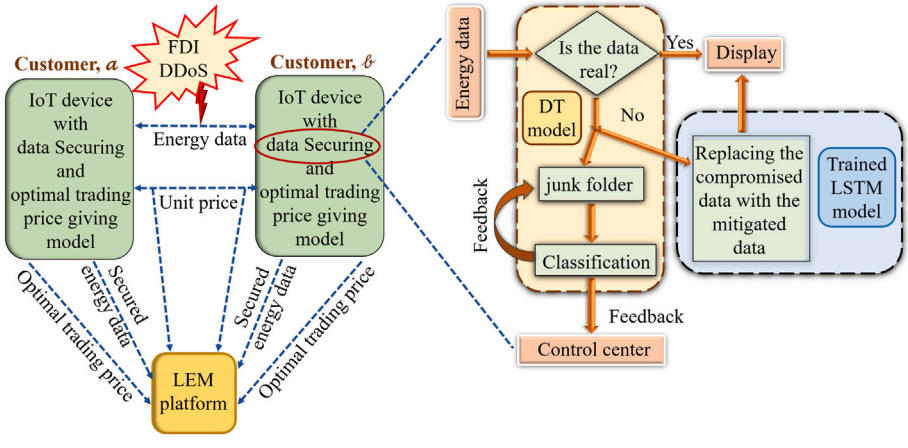


Fig. 2. An illustration of proposed framework for securing P2P energy trade along with optimal trading price.

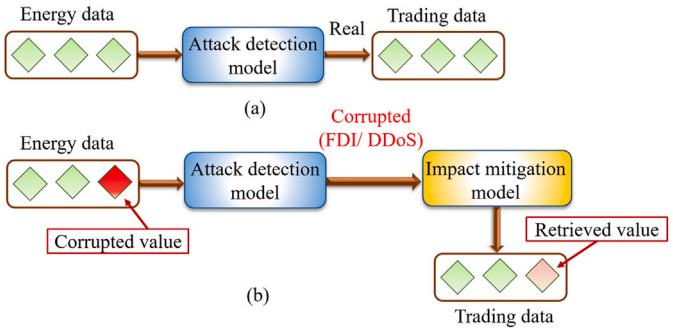


Fig. 3. An illustration of data securing layer.

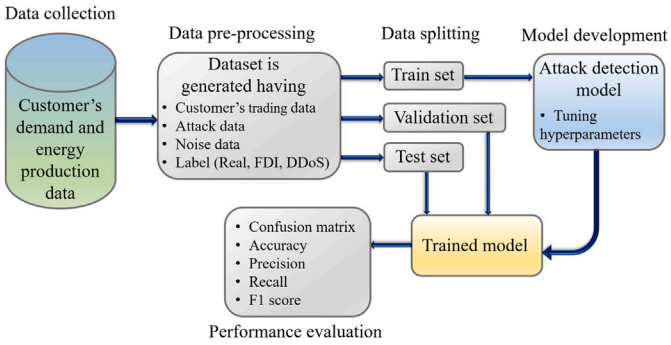


Fig. 4. An illustration of workflow of attack detection model.

- **Mitigation**: Next, predicted net metering values produced by a DL model, LSTM, are used to replace the corrupted data. This ensures that trading operations will continue with the least amount of interruption.
- Following the attack mitigation step, the predicted values are used to trade energy within the LEM. Then the framework uses a novel real-time optimal trading price-giving model, a combination of k-Means clustering and nearest neighbor search algorithm (k-Mc-NnSA), to suggest best price for each trading instant.

The framework's architecture is considered a layered modular design instead of integrating components in a tightly coupled manner. In order to guarantee the following major technical concerns, the proposed scheme for securing trading environment is constructed as the first layer and the optimal trading price-giving model is the second layer of the framework:

- **Layered design**: By separating the two schemes, the framework ensures that the logic and algorithms used for trading price suggestion are neither hampered nor interfered with by attack mitigation. Hence, one scheme can be improved or replaced (for example, by implementing a new forecasting technique or price-giving algorithm) without requiring modifications to the other. This design also encourages scalability.
- **Data stream flow:** The pricing scheme uses the clean, predicted data generated by the mitigation approach as an input. In addition to avoiding redundancy, this sequential but separate flow guarantees that the underlying data securing algorithms have no influence on real-time pricing.

This modular architecture conforms to modern software engineering techniques, such as separation of concerns, and exhibits a design that is both scalable and flexible to future improvements. The following subsections describe the stages of the whole framework in depth.

4.1. Adversarial activity identification model

The development of attack identification model is described in this subsection. The attack detection model's development process is depicted in Fig.4, which includes steps such as data collection, pre-processing, model training, and performance assessment using common classification metrics.

Data about customer demand and energy generation are gathered at the data collection step. Then, a dataset containing the customer's original trading data, corrupted data, noisy data, and data labels identifying the data type (such as Real, FDI, or DDoS) is produced. The dataset is then divided into training, testing and validation sets. By adjusting the hyperparameters, the attack detection model is developed using the training and validation sets. The test set is then used to assess the performance of the trained model through important evaluation metrics, e.g., confusion matrix, accuracy, precision, recall, and F1 score. After comparing the performance evaluation metrics of SVM, M-LP, and DT in this section, the best AI-based model is chosen for the framework's attack detection and classification step.

4.1.1. Attack-inclusive dataset preparation

Let all sets of customers, including those who were victimized, $\mathfrak{z}_{\_v}$, as well as secure $\mathfrak{z}_{\_f}$, be represented by $\mathfrak{z} = \{\mathfrak{z}_1, \mathfrak{z}_2, \ldots, \mathfrak{z}_n\}, \forall \mathfrak{z} \in \mathbb{S}$ such that $\mathfrak{z} = \mathfrak{z}_{\_v} \cup \mathfrak{z}_{\_f}$. $\mathfrak{z}_i$ is the i^{th} customer, and n is the total number of $\mathfrak{z}$, where $(i = 1, 2, \ldots, n)$, and $\mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_i}$ indicates the dataset of i^{th} customer for a time span $\mathcal{E}$. It is assumed that, each customer keeps dataset $\mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_i}$ on their respective device locally. Dataset, $\mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_i}$ has two parts namely: corrupted part, containing the data related to the victim customers, $\mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_{i\_v}}$ and original part, having data related to the safe customers, $\mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_{i\_f}}$, which should follow:

$$\mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_i} = \mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_{i\_v}} \cup \mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_{i\_f}} \tag{8}$$

$$\mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_{i\_v}} = \mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_{i\_di}} \cup \mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_{i\_ddos}} \tag{9}$$

$\mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_{i\_di}}$ and $\mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_{i\_ddos}}$ indicate the corrupted data having FDI and DDoS templates, respectively.

4.1.2. Data processing workflow

This work assumes the appearance of bad data similar to white noise, following a Gaussian distribution pattern.

$$\mathfrak{o}_i \sim \mathcal{O}(\mu, \varrho) \tag{10}$$

and

$$\mathcal{O}_t = \begin{bmatrix} \mathfrak{o}_1 \\ \mathfrak{o}_2 \\ \vdots \\ \mathfrak{o}_n \end{bmatrix} \tag{11}$$

where $\mu = 0$; the mean of the Gaussian distribution. $\varrho = 0.001$; the standard deviation of the Gaussian distribution.

The power spectral density of the noise data $\mathcal{O}_t$ is $\mathcal{D}_{\mathfrak{o}}(\mathcal{F}) = \mu^2$ for all frequencies $\mathcal{F}$. Any pair of $\mathcal{O}_t$'s elements is uncorrelated. For instance, $\mathfrak{o}_1$ and $\mathfrak{o}_2$ are uncorrelated if $t \neq (t+1)$. After considering the outlier data, the noisy trade worthy energy readings become:

$$\mathbb{Q}^{\mathcal{E}'}_{\mathfrak{z}_i} = \begin{bmatrix} \mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_1} + \mathfrak{o}_1 \\ \mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_2} + \mathfrak{o}_2 \\ \vdots \\ \mathbb{Q}^{\mathcal{E}}_{\mathfrak{z}_n} + \mathfrak{o}_n \end{bmatrix} \tag{12}$$

4.1.3. Objective

The trade worthy energy dataset for i^{th}, prosumer, $\mathcal{Q}_{\mathcal{J}_i}^{\mathcal{TR}}$ having $\mathcal{Z}$ classes can be represented as,

$$\mathcal{Q}_{\mathcal{J}_i}^{\mathcal{TR}} = \left\{\mathcal{a}_{\mathcal{j}}, \mathcal{\ell}_{\mathcal{j}}\right\}_{\mathcal{j}=1}^{m} \tag{13}$$

with $\mathcal{\ell} \in \mathcal{L} = \{1, \ldots, \mathcal{Z}\}$ containing its actual labels and $\mathcal{a} \in \mathcal{A} \subset \mathbf{R}$ signifying an example of the attribute. For $\mathcal{j}^{th}$ occurrence, where $(\mathcal{j} = 1, 2, \ldots, m)$, m is total number of occurrences, $\mathcal{a}_{\mathcal{j}}$ is the attribute and its true label is $\mathcal{\ell}_{\mathcal{j}}$. Total data collection $\mathcal{Q}_{\mathcal{J}_i}^{\mathcal{TR}}$ is separated into training set, $\mathcal{Q}_{\mathcal{J}_{i\_tr}}^{\mathcal{TR}}$, testing set $\mathcal{Q}_{\mathcal{J}_{i\_tst}}^{\mathcal{TR}}$, and validation set $\mathcal{Q}_{\mathcal{J}_{i\_v\ell}}^{\mathcal{TR}}$. A function $f(\mathcal{a}, \Pi)$ with values Π that relate the input information to the category, $f : \mathcal{A} \rightarrow \mathcal{L}$, will be learned by the model of classification during its training phase. Parameters Π that make the model's predictions $f(\mathcal{a}_{\mathcal{j}}; \Pi)$ as close as possible to the true labels $\mathcal{\ell}_{\mathcal{j}}$ are the objective of this phase. The objective function during the learning process of the DT model is

$$\min_{\Pi} \frac{1}{m} \sum_{\mathcal{j}=1}^{m} \mathcal{H}(f(\mathcal{a}_{\mathcal{j}}; \Pi), \mathcal{\ell}_{\mathcal{j}}) \tag{14}$$

The goal is to minimize the objective function with respect to the model parameters Π. $\mathcal{H}$ denotes the loss function, which measures the discrepancy between the predicted output $f(\mathcal{a}_{\mathcal{j}}; \Pi)$ and the true label $\mathcal{\ell}_{\mathcal{j}}$. The model is supposed to accurately estimate the benign sample for an experimental sample, i.e., $f(\mathcal{a}_{\mathcal{b}n}; \Pi) = \mathcal{\ell}_{\mathcal{b}n}, \mathcal{a}_{\mathcal{b}n} \in \mathcal{Q}_{\mathcal{J}_{i\_tst}}^{\mathcal{TR}}$. $\mathcal{a}_{\mathcal{b}n}$ represents the original data and its original label is $\mathcal{\ell}_{\mathcal{b}n}$. Additionally, for any input that involves corrupted data, it is preferable to anticipate the adversarial class: $f(\mathcal{a}_{\mathcal{c}m}; \Pi) = \mathcal{\ell}_{\mathcal{c}m}, \mathcal{a}_{\mathcal{c}m} \in \mathcal{Q}_{\mathcal{J}_{i\_tst}}^{\mathcal{TR}}$. Each of the adversary categories is represented by $\mathcal{\ell}_{\mathcal{c}m}$, and $\mathcal{a}_{\mathcal{c}m}$ denotes each of their individual data points

4.1.4. Data splitting

Preprocessed data considering all of the attack patterns of i^{th} customer, $\mathcal{Q}_{\mathcal{J}_i}^{\mathcal{TR}}$, is divided into training, testing and validation sets, represented by $\mathcal{Q}_{\mathcal{J}_{i\_tr}}^{\mathcal{TR}}$, $\mathcal{Q}_{\mathcal{J}_{i\_tst}}^{\mathcal{TR}}$ and $\mathcal{Q}_{\mathcal{J}_{i\_v\ell}}^{\mathcal{TR}}$ respectively, where

$$\mathcal{Q}_{\mathcal{J}_i}^{\mathcal{TR}} = \mathcal{Q}_{\mathcal{J}_{i\_tr}}^{\mathcal{TR}} \cup \mathcal{Q}_{\mathcal{J}_{i\_tst}}^{\mathcal{TR}} \cup \mathcal{Q}_{\mathcal{J}_{i\_v\ell}}^{\mathcal{TR}} \tag{15}$$

The split training, testing, and validation datasets maintain the following relationship with the dataset of i^{th} customer

$$\mathcal{Q}_{\mathcal{J}_{i\_tr}}^{\mathcal{TR}} = \mathcal{x} \times \mathcal{Q}_{\mathcal{J}_i}^{\mathcal{TR}} \tag{16}$$

$$\mathcal{Q}_{\mathcal{J}_{i\_tst}}^{\mathcal{TR}} = \mathcal{y} \times \mathcal{Q}_{\mathcal{J}_i}^{\mathcal{TR}} \tag{17}$$

and

$$\mathcal{Q}_{\mathcal{J}_{i\_v\ell}}^{\mathcal{TR}} = \mathcal{z} \times \mathcal{Q}_{\mathcal{J}_i}^{\mathcal{TR}} \tag{18}$$

while splitting, the values of $\mathcal{x}, \mathcal{y}$ and $\mathcal{z}$ are considered as 0.7, 0.15 and 0.15, respectively.

4.1.5. Support vector machine (SVM)

Support vector machine (SVM) classification models are used here for attack detection because of their efficacy in a variety of domains, including pattern identification, computer vision, visual analysis, and business insight [45,46]. The foundation of SVM, an ML-based technique, is identifying the best hyperplane to divide the different classes in the input data space. It functions by representing each feature as a dimension and encoding incoming data as points in a high-dimensional space. Finding the perfect hyperplane involves addressing an optimization issue as part of SVM training. The goal of this optimization task is to maximize the margin while reducing classification error. This model can categorize new data points according to which side of the hyperplane they belong to after the optimal hyperplane has been identified during training.

Table 3
Architecture of M-LP model for classification task.

Type	Details
Input layer	Single input
Hidden layer	Fully connected layer with 35 neurons
Output layer	3 neurons
Activation function	*tansig* for hidden layer, *softmax* for output

Table 4
Comparison of intrusion detection accuracy of designed SVM, M-LP and DT.

Prosumer	Data set	$\mathcal{A}_{\mathcal{c}}$ of SVM (%)	$\mathcal{A}_{\mathcal{c}}$ of M-LP (%)	$\mathcal{A}_{\mathcal{c}}$ of DT (%)
$\mathcal{J}_5$	Validation set	92.80	92.31	99.01
	Test set	96.77	96.03	98.51
$\mathcal{J}_9$	Validation set	95.29	95.29	98.01
	Test set	94.29	94.04	97.77
$\mathcal{J}_{11}$	Validation set	95.53	92.06	99.01
	Test set	95.04	96.53	97.77
$\mathcal{J}_{13}$	Validation set	94.04	93.80	98.76
	Test set	96.28	94.04	99.26
$\mathcal{J}_{22}$	Validation set	98.51	93.30	99.26
	Test set	98.76	96.77	98.76
$\mathcal{J}_{26}$	Validation set	98.76	95.53	99.01
	Test set	98.26	93.80	99.50

In this work, a multi-class classification model was created utilizing the error-correcting output codes (ECOC) architecture. The ECOC model extends binary SVMs to accommodate multi-class classification challenges. In particular, the One-vs-One coding strategy was used, which trains a distinct binary SVM classifier for each unique pair of class labels. The classification task included three classes: DDoS, FDI, and Real, producing three binary learners. Each binary learner was trained with a linear kernel function. Attack identification performance of the designed SVM for the victim customers, both for the validation and test sets, is enlisted in Table 4.

4.1.6. Multi-layer perceptron (M-LP)

The multi-layer perceptron (M-LP), DL-based algorithm, has demonstrated noteworthy performance in classification tests [47,48], therefore it is adopted here to identify compromised data. The network processes the input data to produce the expected result. The error, or the difference between the expected and actual outputs, is then calculated. The error is then propagated backward through the network modifying the weights and biases. This entails determining the gradient of the error with respect to each weight and bias using the chain rule. To further reduce error, gradient descent is utilized to update the weights and biases. The M-LP model's design architecture is summarized in Table 3. Table 4 lists the M-LP assessment metrics for the validation and test sets for the victim prosumers' attack detection.

4.1.7. Decision tree

Step 1: **Tree design:**

At this stage, DT classifier model is constructed to identify and categorize adversarial instances. The i^{th} customer's learning data set can be expressed as

$$\mathcal{Q}_{\mathcal{J}_{i\_tr}}^{\mathcal{TR}} = [(\mathcal{a}_{\mathcal{j}}, \mathcal{\ell}_{\mathcal{j}}), \ldots, (\mathcal{a}_{m}, \mathcal{\ell}_{m})] \tag{19}$$

Here, $\mathcal{a}_{\mathcal{j}}$ is the attribute and its true label is $\mathcal{\ell}_{\mathcal{j}}$ for $\mathcal{j}^{th}$ occurrence and $\mathcal{\ell} \in \mathcal{L} = \{1, \ldots, \mathcal{Z}\}$; $\mathcal{Z}$ denotes the total number of classes the dataset has. The model uses the Gini impurity to calculate the probability that an observation will be misclassified. For example, if the probability of selecting a data point from class $\mathcal{Z}$ is $\mathcal{C}_{\mathcal{j}}$ and the probability of not selecting one is $(1 - \mathcal{C}_{\mathcal{j}})$, then it can be represented as:

$$\mathcal{I}(\mathcal{Q}_{\mathcal{J}_{i\_tr}}^{\mathcal{TR}}) = \sum_{\mathcal{k}=1}^{\mathcal{Z}} (\mathcal{C}_{\mathcal{j}})(1 - \mathcal{C}_{\mathcal{j}}) \tag{20}$$

It can be further simplified as

$$\mathcal{I}(\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_tr}}) = 1 - \sum_{\mathcal{K}=1}^{\mathcal{Z}} (\mathcal{C}_{\mathcal{J}})^2 \tag{21}$$

In order to effectively detect and classify the adversary instances by the DT classifier the hyperparameters; maximum depth $\mathcal{M}_{x\_\mathcal{D}_p}$, minimum samples per leaf, $\mathcal{M}_{n\_\mathcal{S}_\ell}$, and minimum samples per split $\mathcal{M}_{n\_\mathcal{S}_\mathcal{J}}$; are designed carefully. Moreover, a specific random seed $\mathcal{W}$ is chosen to ensure the repeatability of the split between train and test by controlling the shuffling and splitting of the dataset. A brief about the hyperparameters is as follows:

- $\mathcal{M}_{x\_\mathcal{D}_p}$: This parameter limits the maximum depth of the tree. During the training, $\mathcal{M}_{x\_\mathcal{D}_p} = \infty$ is considered in order for the tree to continue growing until all of its leaves are genuine or until the stopping criteria are met.
- $\mathcal{M}_{n\_\mathcal{S}_\ell}$: This parameter specifies the minimum number of samples that a leaf node must have. If a split results in a leaf with fewer than $\mathcal{M}_{n\_\mathcal{S}_\ell}$ samples, that split is discarded. During training, $\mathcal{M}_{n\_\mathcal{S}_\ell} = 1$ is chosen for effective classification.
- $\mathcal{M}_{n\_\mathcal{S}_\mathcal{J}}$: The lowest number of samples needed to split an internal node is determined by this value. A node will not be split if it results in child nodes with fewer than $\mathcal{M}_{n\_\mathcal{S}_\mathcal{J}}$ samples. The value of $\mathcal{M}_{n\_\mathcal{S}_\mathcal{J}}$ is set at 1 during the training phase.

Following is an example, for better understanding, of splitting and stopping criteria of the designed DT. Let n be a node in the DT and let $\mathcal{N}$ denote the set of samples in node n.

- **Splitting criteria:**
  For a potential split of node n into child nodes n_1 and n_2 If min $(|n_1|, |n_2|) < \mathcal{M}_{n\_\mathcal{S}_\ell}$, the split is discarded.
- **Stopping criteria:**
  If $(|\mathcal{N}|) < \mathcal{M}_{n\_\mathcal{S}_\mathcal{J}}$ or n is pure, then n is a leaf node.

Step 2: **Training and validation:**
After that, the training data set, $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_tr}}$, is fed into the DT classifier to train it using data and their corresponding labels. During this training phase, the DT model identifies relationships and patterns between the characteristics and the target variable. To prevent overfitting, the validation set is used to evaluate the model's performance during training and to adjust hyperparameters. The model is then used to generate predictions with new, unused data once the training and validation phases are completed.

Step 3: **Testing:**
It is necessary to evaluate how well the developed DT model detects attack instances. Comparing the expected and actual labels for a classification problem can help achieve this. More details regarding the performance evaluation are provided in the next section. The following is a pseudo-algorithm regarding the designed DT model.

The comparison of attack identification accuracy, $\mathcal{A}_\mathcal{C}$, for the three AI-based algorithms—SVM, M-LP, and DT—is shown in Table 4. From this table it is evident that, DT performs best for attack detection and categorization purposes. Hence, this work selects DT for attack detection and classification task since it shows more efficient performance when compared to other AI-based approaches.

4.2. Attack impact mitigation model

Upon detecting any compromised net metering data, the framework will replace it with predicted data for that instant. For the purpose of data prediction, this work first analyzes the performance of multi-layer perceptron (M-LP) and long short-term memory (LSTM) based schemes.

Algorithm 1 Decision tree classifier.

Input: Preprocessed data $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_i}$
Data splitting:
• $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i,\text{tr}}} \leftarrow 70\ \%$ of $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_i}$
• $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i,\text{tst}}} \leftarrow 15\ \%$ of $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_i}$
• $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i,\text{vl}}} \leftarrow 15\ \%$ of $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_i}$
Output: Training set $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i,\text{tr}}}$,
Testing set $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i,\text{tst}}}$,
Validation set $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i,\text{vl}}}$
Design of DT:
Input: Training set $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i,\text{tr}}}$
Initialize:
• $\mathcal{M}_{x,\mathcal{D}_p} \leftarrow \infty$
• $\mathcal{M}_{n,\mathcal{S}_\ell} \leftarrow 1$
• $\mathcal{M}_{n,\mathcal{S}_\mathcal{J}} \leftarrow 1$
• $\mathcal{I}(\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i,\text{tr}}}) \leftarrow 1 - \sum_{\mathcal{K}=1}^{\mathcal{Z}} (\mathcal{C}_{\mathcal{J}})^2$
Splitting criteria:
Let $\mathcal{N}$ be the set of samples in node n. A potential split into child nodes n_1 and n_2 is:
if $\min(|n_1|, |n_2|) < \mathcal{M}_{n,\mathcal{S}_\ell}$ **then**
  Discard the split
Stopping criteria:
if $|\mathcal{N}| < \mathcal{M}_{n,\mathcal{S}_\mathcal{J}}$ *or node* n *is pure* **then**
  Mark n as a leaf node
Training:
if $|\mathcal{N}| < \mathcal{M}_{n,\mathcal{S}_\mathcal{J}}$ *or* $\mathcal{N}$ *is pure* **then**
  Mark $\mathcal{N}$ as a leaf node
  return $\mathcal{N}$
Best split point: $\mathcal{N}_1, \mathcal{N}_2 \leftarrow$ SplitNode($\mathcal{N}$)
if *Best split is valid* **then**
  $\mathcal{N}_{\text{left}} \leftarrow$ TrainTree($\mathcal{N}_1$)
  $\mathcal{N}_{\text{right}} \leftarrow$ TrainTree($\mathcal{N}_2$)
return node with split points and child nodes $\mathcal{N}_{\text{left}}, \mathcal{N}_{\text{right}}$
Output: Designed DT classifier

Then, a performance comparison of designed M-LP and LSTM based on evaluation metrics will be conducted to select the optimal model for the data prediction purpose.

4.2.1. Preparing dataset

For i^{th} participant, the time series form of SM data containing only the benign trade-worthy data along with outliers can be represented as:

$$\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_i}} = \{\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_1}}, \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_2}}, \dots, \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_m}}\} \tag{22}$$

For sequential length , ℓ, input matrix, $\mathcal{J}$ and the final predicted output vector , $\mathcal{V}$, can be represented as:

$$\mathcal{J}_u = \{\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_u}}, \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{u+1}}}, \dots, \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{u+\ell-1}}}\} \tag{23}$$

and

$$\mathcal{V} = \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{u+\ell}}} \tag{24}$$

for $u = 1, 2, \dots, (m - \ell)$.

4.2.2. Data splitting

Benign trade-worthy data along with outlier data of i^{th} customer, $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_i}}$, is divided into training, testing, and validation

Table 5
Architecture of M-LP model.

Type	Details
Input layer	30 sequence length
Hidden layer	Fully connected layer with 35 neurons
Output layer	Single neuron
Activation function	Log-sigmoid for hidden layer, linear for output

sets, represented by $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{itr}}}$, $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{its\mathcal{t}}}}$, and $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{iv\ell}}}$, respectively, where

$$\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_i}} = \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{itr}}} \cup \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{its\mathcal{t}}}} \cup \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{iv\ell}}} \tag{25}$$

The split training, testing, and validation datasets maintain the following relationship with the dataset of i^{th} customer

$$\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{itr}}} = x \times \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_i}} \tag{26}$$

$$\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{its\mathcal{t}}}} = y \times \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_i}} \tag{27}$$

and

$$\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_{iv\ell}}} = z \times \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_i}} \tag{28}$$

where, the values of x, y, and z are considered as $0.7, 0.15$, and 0.15, respectively.

4.2.3. Multi-layer perceptron (M-LP)

M-LP, a simple form of feed-forward neural network, is utilized here to investigate how well it may lessen the impact of compromised data because it has demonstrated exceptional performance in time-dependent data predictions [49]. To obtain the predicted output, the network processes the input data. Then, the error, which is the discrepancy between the intended output and the actual output, is computed. The weights and biases are subsequently modified by propagating the error backward through the network. Using the chain rule, this involves calculating the gradient of the error with respect to each weight and bias. Furthermore, gradient descent is employed to update the biases and weights in order to reduce error. Table 5 summarizes the design architecture of the M-LP model.

4.2.4. Long short-term memory (LSTM)

A specific kind of RNN called LSTM is designed to model temporal and sequential data [50]. By employing memory cells and gating techniques to capture temporal patterns and long-term dependencies, it overcomes the drawbacks of conventional RNNs. This makes LSTM suitable for time series forecasting, especially when the data is non-linear or reveals complex relationships over time. The detailed modeling of the designed LSTM is given below.

Objective:

The objective of the model is to find the optimal weights $\mathcal{W}$ and biases $\mathcal{B}$ that minimize the error between the predicted output $\mathcal{V}'$ and the actual target $\mathcal{V}$ by considering mean squared error, $\mathcal{E}_{m\mathcal{J}}$. For y number of samples, it can be represented as:

$$\mathcal{E}_{m\mathcal{J}} = \frac{1}{y}\sum_{u=1}^{y}(\mathcal{V} - \mathcal{V}')^2 \tag{29}$$

In order to prevent the issues of vanishing and exploding gradients, the main objective of weight initialization is to guarantee that the gradients employed during the training process are neither too tiny nor too large. Generally, biases are initialized to either zero or slightly positive values. For LSTM modeling purposes, input and recurrent weights are initialized by Xavier Uniform and Orthogonal Initialization techniques, respectively, whereas biases are initialized to zero.

Architecture:

- **Input Layer:** At first, input layer receives a sequence of ℓ time steps: $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_i}} = \{\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_1}}, \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_2}}, \ldots, \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_{i\_\mathcal{B}_\ell}}\} \in \mathbf{R}^{\ell}$
- **LSTM layer:** The operations of the designed LSTM layer having weights and biases $\mathcal{W}_i, \mathcal{W}_o, \mathcal{W}_f, \mathcal{W}_c$ and $\mathcal{B}_i, \mathcal{B}_o, \mathcal{B}_f, \mathcal{B}_c$ for input, output, forget and cell state, respectively, can be described in following manner.

  Considering $\mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_t}$ as the the input vector at the current time step t, the output of the forget gate, $\mathcal{V}_f$ is:

$$\mathcal{V}_f = \sigma(\mathcal{W}_f \cdot \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_t} + \mathcal{B}_f) \tag{30}$$

Which elements of the prior cell state should be kept and which should be discarded are decided by the forget gate. Here, σ is the sigmoid activation function. Which new data should be stored in the cell state is determined by the input gate. The output of this gate can be represented as:

$$\mathcal{V}_i = \sigma(\mathcal{W}_i \cdot \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_t} + \mathcal{B}_i) \tag{31}$$

Candidate values $\widetilde{\mathcal{C}_t}$ that could be added to the cell state are calculated by the cell state update as follows:

$$\widetilde{\mathcal{C}_t} = \tanh(\mathcal{W}_c \cdot \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_t} + \mathcal{B}_c) \tag{32}$$

Now the scaled previous cell state and the new candidate values are combined to update the cell state by following

$$\mathcal{C}_t = \mathcal{V}_f \odot \mathcal{C}_{t-1} + \mathcal{V}_i \odot \widetilde{\mathcal{C}_t} \tag{33}$$

The output gate controls the proportion of the updated cell state, $\mathcal{C}_t$ that goes into the hidden state $\mathcal{H}_t$ by following:

$$\mathcal{V}' = \sigma(\mathcal{W}_o \cdot \mathcal{Q}^{\mathcal{TR}}_{\mathcal{J}_t} + \mathcal{B}_o) \tag{34}$$

Here, $\mathcal{V}'$ is the predicted output. By using the output gate to modulate the updated cell state $\mathcal{C}_t$, the hidden state $\mathcal{H}_t$ is calculated as follows:

$$\mathcal{H}_t = \mathcal{V}' \odot \tanh(\mathcal{C}_t) \tag{35}$$

- **Fully connected layer:** In this layer weighted sum of hidden states is calculated using a linear activation function, ρ as shown below:

$$\mathcal{V}' = \rho(\mathcal{W}_o \cdot \mathcal{H}_t + \mathcal{B}_o) \tag{36}$$

- **Output layer:** It produces the final predicted output $\mathcal{V}'$.

  Next, the Adam optimization process is used to update the weights and biases in the following manner, having ζ as the learning rate:

$$\mathcal{W} \leftarrow \mathcal{W} - \zeta\frac{\partial \mathcal{E}_{m\mathcal{J}}}{\partial \mathcal{W}} \tag{37}$$

$$\mathcal{B} \leftarrow \mathcal{B} - \zeta\frac{\partial \mathcal{E}_{m\mathcal{J}}}{\partial \mathcal{B}} \tag{38}$$

In this case, $\mathcal{W}$ and $\mathcal{B}$ stand for the general weights and biases respectively of the hidden and output layers. Table 6 summarizes the modeling architecture of LSTM based forecasting model.

To assess the effectiveness of the designed M-LP and LSTM in retrieving original trading data, two commonly utilized evaluation metrics associated with forecasting models, mean absolute error ($\mathcal{E}_{\mathcal{MA}}$) and root mean squared error ($\mathcal{E}_{\mathcal{RMS}}$), are used. For the victim customers, Table 7 lists the numerical values of $\mathcal{E}_{\mathcal{MA}}$ and $\mathcal{E}_{\mathcal{RMS}}$ for the M-LP and LSTM.

The table demonstrates that, LSTM consistently outperforms M-LP for all target customers when evaluated using $\mathcal{E}_{\mathcal{MA}}$. While M-LP produces better results than LSTM in terms of $\mathcal{E}_{\mathcal{RMS}}$ for two customers—$\mathcal{J}_{13}$ and $\mathcal{J}_{22}$—LSTM performs better considering overall situation. Since LSTM outperforms M-LP in the performance metrics investigation shown in Table 7, this work uses LSTM to predict trade-worthy energy data after identifying any compromised event.

Table 6
Architecture of LSTM model.

Type	Details
Input size	Single dimension
Hidden layer 1	LSTM layer with 128 units, outputting sequences.
Hidden layer 2	LSTM layer with 256 units, outputting a single value
Output layer	One fully connected layer to predict a single value
Regression layer	Final layer for continuous output prediction
Learning rate	0.00125
Mini-batch size	32 samples
Epoch	150

Table 7
Comparison of data forecasting ability between M-LP and LSTM.

Identity of customer	$\mathcal{E}_{\mathcal{MA}}$ of LSTM	$\mathcal{E}_{\mathcal{MA}}$ of M-LP	$\mathcal{E}_{\mathcal{RMS}}$ of LSTM	$\mathcal{E}_{\mathcal{RMS}}$ of M-LP
$\mathcal{S}_5$	0.089	0.106	0.206	0.226
$\mathcal{S}_9$	0.064	0.077	0.114	0.122
$\mathcal{S}_{11}$	0.082	0.137	0.163	0.195
$\mathcal{S}_{13}$	0.087	0.101	0.176	0.138
$\mathcal{S}_{22}$	0.061	0.079	0.116	0.107
$\mathcal{S}_{26}$	0.094	0.155	0.269	0.334
Average	0.081	0.109	0.17	0.187

4.3. Optimizing trading price

This section analyzes the performance of random forest (RF) and a new scheme based on k-means clustering integrated with nearest neighbor search algorithm (k-Mc-NnSA) in providing optimal trading prices in a comparative way. Then, based on the performance analysis, a more suitable model is selected for the proposed framework.

4.3.1. Preparing dataset

The price data offered by seller and buyer during energy trade in ¢/kWh can be represented respectively as

$$\mathcal{S}^{\mathcal{TP}} = \begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} & \cdots & \mathcal{S}_{1t} \\ \mathcal{S}_{21} & \mathcal{S}_{22} & \cdots & \mathcal{S}_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{S}_{d1} & \mathcal{S}_{d2} & \cdots & \mathcal{S}_{dt} \end{bmatrix} \tag{39}$$

and

$$\mathcal{B}^{\mathcal{TP}} = \begin{bmatrix} \mathcal{B}_{11} & \mathcal{B}_{12} & \cdots & \mathcal{B}_{1t} \\ \mathcal{B}_{21} & \mathcal{B}_{22} & \cdots & \mathcal{B}_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{B}_{d1} & \mathcal{B}_{d2} & \cdots & \mathcal{B}_{dt} \end{bmatrix} \tag{40}$$

Here d is the number of data points, i.e., number of sellers or buyers, and t represents the different instances of transactions. Then, the two price data sets are combined as follows to train the price-suggesting model.

$$\mathcal{P} = \left[\mathcal{S}^{\mathcal{TP}}, \quad \mathcal{B}^{\mathcal{TP}}\right] \tag{41}$$

or

$$\mathcal{P} = \begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} & \cdots & \mathcal{S}_{1t} & \mathcal{B}_{11} & \mathcal{B}_{12} & \cdots & \mathcal{B}_{1t} \\ \mathcal{S}_{21} & \mathcal{S}_{22} & \cdots & \mathcal{S}_{2t} & \mathcal{B}_{21} & \mathcal{B}_{22} & \cdots & \mathcal{B}_{2t} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{S}_{d1} & \mathcal{S}_{d2} & \cdots & \mathcal{S}_{dt} & \mathcal{B}_{d1} & \mathcal{B}_{d2} & \cdots & \mathcal{B}_{dt} \end{bmatrix} \tag{42}$$

4.3.2. Random forest (RF)

An ensemble learning system called random forest (RF) aggregates the results of several decision trees to decrease overfitting and increase prediction accuracy [51]. Using bootstrapped subsets of the data (sampling with replacement), the algorithm generates numerous decision trees during training. To ensure variation among the trees, a subset of features is randomly chosen at each split. Without pruning, each tree is grown independently to its maximum depth. The final prediction for regression tasks is determined by averaging the outputs of all trees. Because it preserves complex, non-linear relationships in the data and minimizes overfitting when compared to a single decision tree, this method makes RF robust. By assessing the relative contributions of each feature to the model's predictions, it also offers insights into feature importance, making it a flexible and dependable tool for a range of ML tasks. Table 8 provides the architecture of the used RF model for determining optimal trading prices.

Table 8
Architecture of RF model.

Type	**Details**
Input features	Seller price (¢/kWh), Buyer price (¢/kWh)
Target variable	Optimal trading price
Model	RF Regressor with 50 DTs trained using bootstrapped subsets of the data.
Feature sampling	Random subset of features at each split
Training	Each tree learns the relationship between the input features and the target variable
Output	Suggested optimal trading price, $\mathcal{TP}_o$ for new seller-buyer price pairs

4.3.3. ANN-based approximate nearest neighbor (ANN) search algorithm

The application of artificial neural networks (ANNs) in various areas is quickly rising these days. On the other hand an approximate nearest neighbor search technique aims to quickly find the closest neighbors of a query point [52]. An ANN-based approximate nearest neighbor (ANN) search algorithm is developed here to offer optimal trading prices during consumers' energy transactions. The network architecture comprises an input layer, one hidden layer, and an output layer designed for generating optimal trading prices. The input layer accepts two features: price data offered by seller, $\mathcal{S}^{\mathcal{TP}}$, and buyer, $\mathcal{B}^{\mathcal{TP}}$, at a given instant. These features are transmitted through a hidden layer made up of 10 fully connected neurons utilizing the hyperbolic tangent sigmoid activation function, allowing the network to capture complicated, non-linear interactions between inputs and outputs. The final output layer consists of a single neuron with a linear activation function that produces an optimal trading price. Table 9 shows the architecture of the implemented ANN model used to determine optimal trading prices.

4.3.4. k-means clustering with nearest neighbor search algorithm (k-mc-NnSA)

A novel scheme combining k-means clustering (k-Mc) and nearest neighbor search algorithm (k-Mc-NnSA), for suggesting optimal trading price is adopted here. k-Mc algorithm is commonly used for clustering tasks due to its low computational complexity and ease of implementation [53,54]. This research uses the k-Mc in a unique way to facilitate LEM trading. At first, this work uses k-Mc as a technique for efficient customer segmentation. Then, for finding an optimal trading price, $\mathcal{TP}_o$, for a trading instant, t, an approach based on the combined architecture of k-Mc and nearest neighbor search algorithm (NnSA) is used. Primarily from historical data, the k-Mc model is trained, and after that, for a trading instance, using the previously trained model, the optimal trading price is determined with the help of NnSA. By assembling comparable trade prices of seller and buyer into discrete clusters that reflect various market circumstances or price categories, the designed k-Mc-NnSA fulfills the following purposes:

Table 9
Architecture of the ANN model.

Component	Hyperparameter	Description
Input layer	Number of inputs	2 (Seller price (¢/kWh), buyer price (¢/kWh))
Hidden layer	Number of hidden layers	1
	Number of neurons	10
	Activation function	Hyperbolic tangent sigmoid
Output layer	Number of output neurons	1
	Activation function	Linear activation
Training settings	Training algorithm	Levenberg–Marquardt
	Maximum epochs	1000
Model Output	–	Suggested optimal trading price, $\mathcal{TP}_o$ for new seller–buyer price pairs

- Clustering by k-Mc makes it possible to divide the market into various price ranges, which aids in understanding the diverse tastes and behaviors of buyers and sellers.
- Grouping of comparable prices determines each cluster's average trading price. This helps in avoiding overpricing or underpricing.
- Buyers can save money by taking advantage of potentially reduced prices within a cluster that matches their priceexpectations.
- Targeting clusters where they can optimize their profits without pricing themselves out of the market can be advantageous for sellers.
- To guarantee that trades are carried out at the best prices, NnSA rapidly determines the most suitable trading price for a particular seller's or buyer's price.
- Trading becomes more efficient as a result of the instant price matching, which reduces the time and effort needed for pricing negotiations.

Modeling

- **Objective:** The goal of k-Mc model is to partition the $\mathcal{d}$ data points into $\mathcal{k}$ clusters in such a way that the within-cluster sum of squares is minimized. The objective function during training phase of k-Mc can be represented as:

$$\mathcal{G} = \sum_{i=1}^{\mathcal{k}} \sum_{\mathcal{j}=1}^{\mathcal{d}_i} \|\mathcal{D}_{\mathcal{j}}^{(i)} - \Theta_i\|^2 \tag{43}$$

Here $\mathcal{k}$ is the number of clusters; $\mathcal{D}_{\mathcal{j}}^{(i)}$ is the $\mathcal{j}^{th}$ data point in cluster i; Θ_i is the centroid of cluster i; $\mathcal{d}_i$ is the number of data points in cluster i. This objective function, $\mathcal{G}$, measures the total squared distance between each data point and the centroid of its assigned cluster. The aim is to find the centroids, Θ_i, that minimize this sum.

- **Optimal** $\mathcal{k}$: For choosing an optimal value of $\mathcal{k}$, the elbow-plot method is analyzed here. This plot is an illustration used to calculate the optimal number of clusters $\mathcal{k}$ in k-Mc. It depicts the sum of squared distances (within-cluster sum of squares, also known as inertia) from each point to the cluster centroid against various $\mathcal{k}$ values. The objective is to find the point where increasing $\mathcal{k}$ no longer significantly reduces inertia, resulting in an "elbow". From Fig. 5, it is noticeable that for the optimal price-giving model, the elbow point occurs when $\mathcal{k} = 5$.

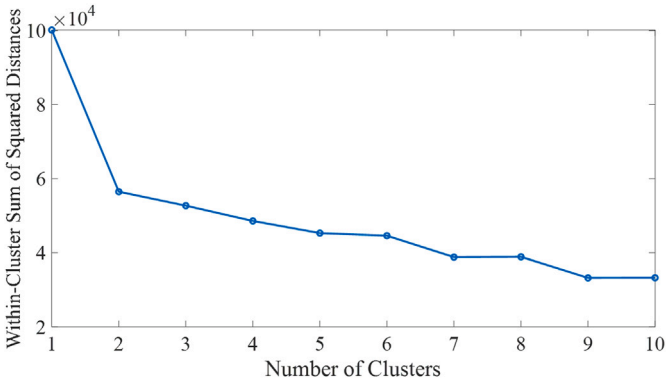


Fig. 5. Elbow plot for k-Mc.

- **Initialization:** To improve the initialization step of the k-Mc to speed up convergence and improve the quality of the final clusters, the K-Means++ algorithm is used here. Let us consider that $\mathcal{C}$ represents the set of all initial centroids as $\mathcal{C} = c_1, c_2, \ldots, c_{\mathcal{k}}$. The first centroid is chosen randomly as: $c_1 = \mathcal{p}_i$ for randomly chosen $i \in 1, 2, \ldots, \mathcal{r}$, where $\mathcal{r}$ is the total number of rows in $\mathcal{P}$ and $\mathcal{p}_i \in \mathcal{P}$.

Then, the distance for each data point, $\mathcal{p}_i \in \mathcal{P}$, to the nearest centroid is calculated as follows:

$$\mathcal{L}(\mathcal{p}_i) = \min_{c \in \{c_1, c_2, \ldots, c_{\mathcal{k}}\}} \|\mathcal{p}_i - c\|^2 \tag{44}$$

Then, the next centroid is selected with a probability proportional to the squared distance:

$$\Phi(\mathcal{p}_i) = \frac{\mathcal{L}(\mathcal{p}_i)^2}{\sum_{\mathcal{p} \in \mathcal{P}} \mathcal{L}(\mathcal{P})^2} \tag{45}$$

This process is repeated until $\mathcal{k}$ centroids are chosen.

- **k-Mc performing:** Then, k-Mc is performed on $\mathcal{P}$ as in (42) to partition the trading price data into $\mathcal{k}$ clusters whose centroids can be denoted as $c_1, c_2, \ldots, c_{\mathcal{k}}$. Each centroid c_i represents the mean of all data points in cluster i as:

$$c_i = \frac{1}{\mathcal{d}_i} \sum_{\mathcal{P} \in \mathcal{d}\mathcal{s}_i} \mathcal{P} \tag{46}$$

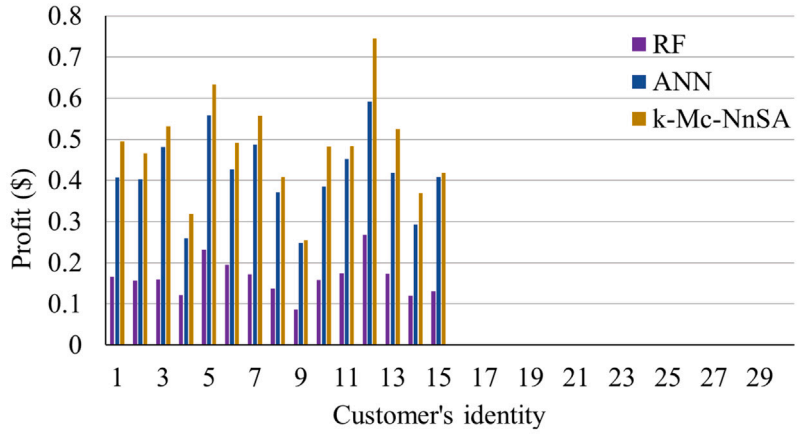
Here $\mathcal{d}\mathcal{s}_i$ is the set of data points assigned to cluster i, and $\mathcal{d}_i$ is the number of data points in cluster i. After obtaining the centroids, the average trading price for each cluster is calculated as:

$$\mathcal{TP}_i = \frac{1}{\ell} \sum_{\mathcal{j}=1}^{\ell} c_{i\mathcal{j}} \tag{47}$$

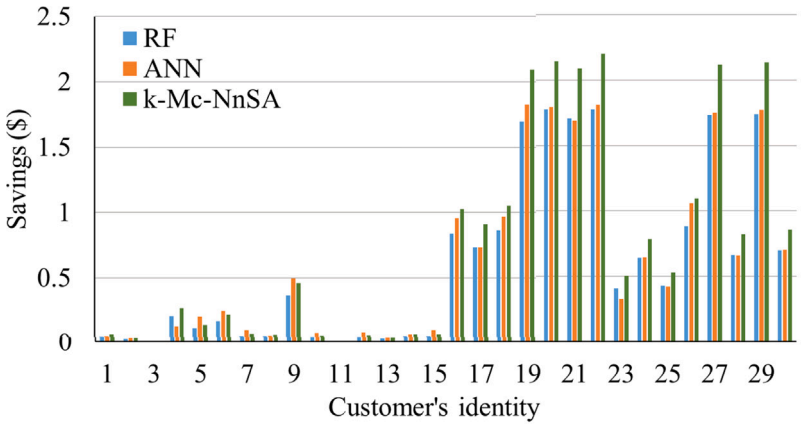
If c_i is the centroid of cluster i, and it is a vector of length ℓ; each element of which represents a price dimension of seller and buyer; the average trading price is represented by (47). Here, $c_{i\mathcal{j}}$ is the $\mathcal{j}^{th}$ component of the i^{th} centroid.

- **Finding optimal trading price:**

After obtaining the centroids c_i, and trading prices, $\mathcal{TP}_i$, the optimal trading price can be determined by applying NnSA for



(a) Profit earned by the seller



(b) Buyer's savings

Fig. 6. Compared trading response of RF, ANN and k-Mc-NnSA.

a transaction instant, t. For a feature vector, $\mathcal{U}$—containing both the seller and buyer prices—the closest centroid ($\mathcal{C}_c$) to $\mathcal{U}$ is first identified by applying Euclidean distance as follows:

$$\mathcal{C}_c = \arg\min_i \|\mathcal{U} - c_i\| \tag{48}$$

where $\|\mathcal{U} - c_i\|$ is the Euclidean distance between $\mathcal{U}$ and c_i.

Then, the optimal trading price, $\mathcal{TP}_o$, for the current transaction, say a^{th}, between seller x and buyer z, is set to the trading price of the closest centroid as:

$$\mathcal{TP}_o(a \mid x, z) = \mathcal{TP}(\mathcal{C}_c) \tag{49}$$

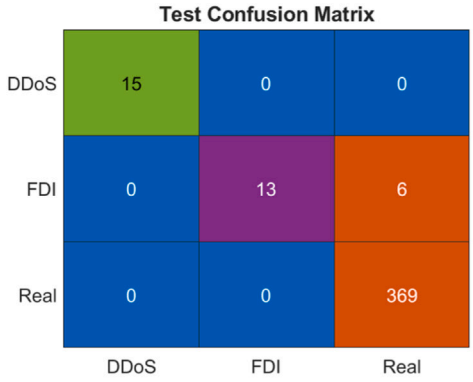
The objective of using k-Mc here is to facilitate the optimal price-finding technique, not to evaluate how effectively it can cluster the price data of sellers and buyers. That's why the clustering-related evaluation metrics, e.g., silhouette score, are not analyzed here.

Through financial benefit analysis of seller and buyer, the effectiveness of proposed schemes, RF, ANN, and k-Mc-NnSA, is justified. Fig. 6 illustrates the financial benefit of the considered LEM structure for both sellers and buyers over a 24 hour time horizon when RF, ANN, and k-Mc-NnSA are used for suggesting $\mathcal{TP}_o$. Fig. 6 shows the profit (in $) earned by the sellers, $\mathcal{P}_s$, whereas Fig. 6 depicts buyers' savings, $\mathcal{S}_b$, (in $) for the assumed time frame. Having zero benefit in Fig. 6a and b indicates that, at that instant no trading is performed by that customer. While comparing these three models— RF, ANN, k-Mc-NnSA—market structure, market conditions and customers' price data are kept the same. The graphical analysis of $\mathcal{P}_s$ and $\mathcal{S}_b$ in Fig. 6 reveals that the developed k-Mc-NnSA surpasses RF and ANN in maintaining the financial benefit of both buyers and sellers. Fig.4a indicates that proposed k-Mc-NnSA model consistently shows higher profit for the seller across all trading instances compared to RF and ANN. This model again outperforms both RF and ANN in maintaining buyers' savings for all the trading events as depicted by Fig.4b. Table 10 enlists the convergence time of the optimal trading price giving models for the assumed market structure over 24 hour time horizon. The table shows that, the k-Mc-NnSA model has the fastest convergence time (175.45 s) among the evaluated approaches, making it appropriate for real-time optimal trading price estimation in LEM. Hence, this work chooses k-Mc-NnSA model for obtaining $\mathcal{TP}_o$ and it will help to encourage customers to participate in P2P trading.

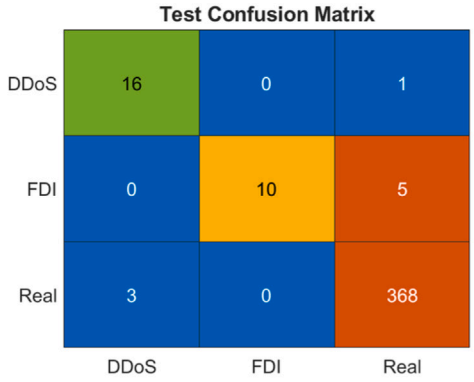
Table 10
Model convergence times for optimal trading price estimation.

Models	Execution time
ANN	476.65 s
RF	344.21 s
k-Mc-NnSA	175.00 s

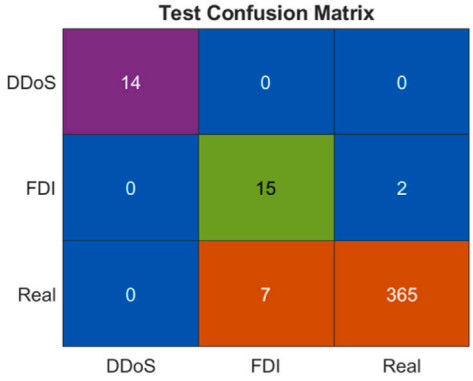
The term "AI-based" in these working steps refers to the employment of unsupervised ML (k-Mc) and an NnSA to achieve real-time optimal pricing in a LEM. While the components (k-Mc and NnSA) are traditional, their combined application allows the system to autonomously learn market dynamics and make intelligent, data-driven pricing decisions without any operator involvement.



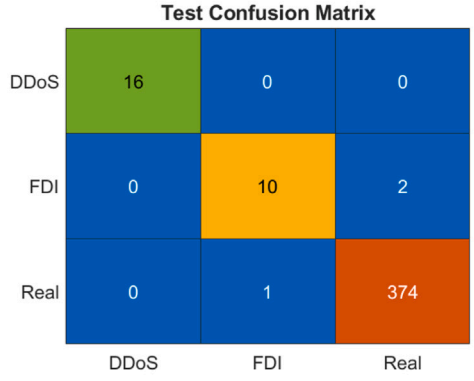
(a) Confusion matrix of $\mathcal{J}_5$ for test set.



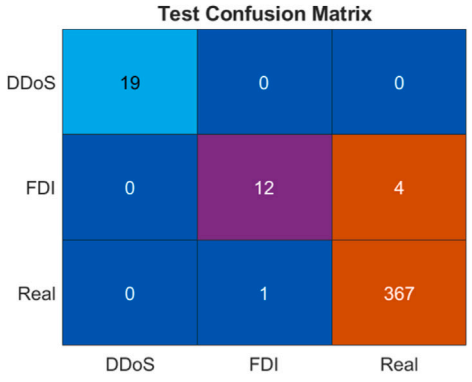
(b) Confusion matrix of $\mathcal{J}_9$ for test set.



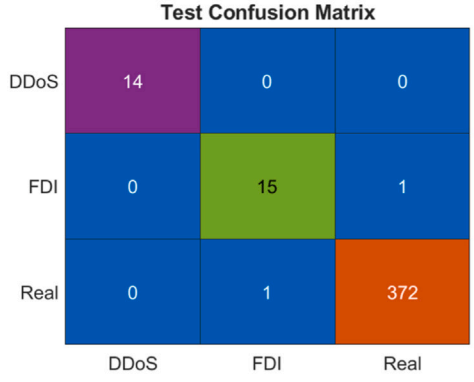
(c) Confusion matrix of $\mathcal{J}_{11}$ for test set.



(d) Confusion matrix of $\mathcal{J}_{13}$ for test set.



(e) Confusion matrix of $\mathcal{J}_{22}$ customer for test set.



(f) Confusion matrix of $\mathcal{J}_{26}$ for test set.

Fig. 7. Visual representation of confusion matrices of victim prosumers for test set.

5. Performance analysis

This part includes the simulation results and a detailed analysis of the designed framework's performance in safeguarding the P2P energy transaction, as well as the impact of the optimal trading price model on seller and buyer financial benefits.

5.1. Experimental set-up

The framework is designed and executed within the MATLAB 2021 a environment, using MATLAB's computational capabilities. The computer is equipped with an Intel(R) Core(TM) i5-8250U CPU with a clock speed of 1.60 to 1.80 GHz, 4 GB of RAM, and a 64-bit version of Windows 10 Home Single Language. With this configuration, the corrupted data identification and retrieval of original data through forecasting are performed instantly for a single value. While using k-Mc-NnSA for providing $\mathcal{TP}_{\sigma}$, it takes approximately 175 s to execute the considered LEM structure over a time span of 24 hours.

5.2. Dataset

The raw dataset includes the energy demand of all customers and energy generated by them through RESs for a time horizon of 7 days at an interval of 5 minutes. Then, following (1), the trade-worthy net energy data is calculated. This dataset, without having any compromised portion, is used to train, validate and evaluate the LSTM-based forecasting model. In order to train the proposed attack identifying model, DT-based model, a dataset is prepared by maintaining the attack strategy as in Table 2 and templates as given in (5) and (7). Along with the net metering data, buyer and seller price data are used to evaluate the

Table 11
Intrusion detection accuracy of designed DT.

Identity of prosumer	For validation set (%)	For test set (%)
$\mathfrak{I}_5$	99.01	98.51
$\mathfrak{I}_9$	98.01	97.77
$\mathfrak{I}_{11}$	99.01	97.77
$\mathfrak{I}_{13}$	98.76	99.26
$\mathfrak{I}_{22}$	99.26	98.76
$\mathfrak{I}_{26}$	99.01	99.50

Table 12
Measures for assessing the integrity of data.

Identity of prosumer's	$\mathcal{P}_{on}$(%)	$\mathcal{R}_{c\ell}$(%)	$\mathcal{S}_{\mathcal{F}1}$(%)
5th	97.47	89.47	93.48
9th	94.20	86.66	89.23
11th	89.21	95.45	91.90
13th	96.79	94.36	95.52
22nd	97.08	91.58	94.03
26th	97.83	97.83	97.83

Here, $\mathcal{P}_{on}$, $\mathcal{R}_{c\ell}$, and $\mathcal{S}_{\mathcal{F}1}$ indicate the precision, recall and F1 score, respectively.

performance of k-Mc-NnSA for providing optimal trading prices for a single day.

5.3. Robustness

As dividing a dataset into training, testing, and validation sets is essential for building a robust ML model, this work utilizes 70 %, 15 %, and 15 % of the data for training, validation, and testing purposes, respectively. The training set, which is the core component, lowers the chance of underfitting by enabling the model to learn patterns efficiently. In order to avoid overfitting, the validation set is used to evaluate the model's performance during training and adjust hyperparameters. Lastly, the testing set ensures the model's generalization capabilities by offering a fair assessment of its performance on unknown data. This division prevents data leaks and guarantees a dependable and useful method for developing and assessing ML models.

5.4. Evaluation metrics for attack identifier and classifier

5.4.1. Confusion matrix

Analyzing a confusion matrix is very effective for assessing the performance of a classifier for a batch of data because its innate ability allows one to have an overview of classification performance as a tabular representation. Fig. 7a—f illustrate the confusion matrices of victim customers—$\mathfrak{I}_5, \mathfrak{I}_9, \mathfrak{I}_{11}, \mathfrak{I}_{13}, \mathfrak{I}_{22}$ and $\mathfrak{I}_{26}$—for test dataset. For example, the confusion matrix for $\mathfrak{I}_9$, illustrated in Fig. 7, exhibits the model's effectiveness in identifying real and corrupted data with a very few misclassifications. Investigation of other confusion matrices reveals the same. During the training of the proposed DT model with different customer data sets, the model architecture remains unchanged and it is evident from Fig. 7 that, the proposed DT-based intrusion detection scheme performs effectively for all the victim participants.

5.4.2. Accuracy

To evaluate the performance of a multiclass classification model, the accuracy of the method is often used. To assess the correctness of the trained model, the accuracy $\mathcal{A}_{\mathcal{C}}$, as stated in (50), is calculated by comparing the total number of correct predictions $\mathcal{P}_{\mathcal{C}}$ with the total number of predictions $\mathcal{P}_{\mathcal{T}}$.

$$\mathcal{A}_{\mathcal{C}} = \frac{\mathcal{P}_{\mathcal{C}}}{\mathcal{P}_{\mathcal{T}}} \tag{50}$$

$\mathcal{A}_{\mathcal{C}}$ is represented by %. For instance, if the confusion matrix of Fig. 7 is considered, which displays the performance of the trained DT model for the 5th customer for the test data set, then $\mathcal{P}_{\mathcal{C}}$ and $\mathcal{P}_{\mathcal{T}}$ are 397 and 403, respectively, and thus $\mathcal{A}_{\mathcal{C}}$ is 98.51 % which is enlisted in Table 11. This table also shows how well the designed DT model detected cyberattacks for both the test and validation sets related to the victim customers. The table indicates that, the model works consistently well for all victims, with validation and test accuracies ranging from 97.77 % to 99.50 %, indicating robustness of the designed model.

5.4.3. Precision and recall

While recall ($\mathcal{R}_{c\ell}$) measures the proportion of real positives to all actual positives, precision ($\mathcal{P}_{on}$) measures the proportion of genuine positives to all predicted positives. $\mathcal{P}_{on}$ and $\mathcal{R}_{c\ell}$ can be expressed as follows if $\mathcal{T}_{\mathcal{P}}$ or true positive denotes occurrences that were correctly anticipated to be positive, $\mathcal{F}_{\mathcal{P}}$ or false positive denotes instances that were wrongly forecasted to be positive, and $\mathcal{F}_{\mathcal{N}}$ or false negative denotes instances that were incorrectly estimated to be negative.

$$\mathcal{R}_{c\ell} = \frac{\mathcal{T}_{\mathcal{P}}}{\mathcal{T}_{\mathcal{P}} + \mathcal{F}_{\mathcal{N}}} \tag{51}$$

$$\mathcal{P}_{on} = \frac{\mathcal{T}_{\mathcal{P}}}{\mathcal{T}_{\mathcal{P}} + \mathcal{F}_{\mathcal{P}}} \tag{52}$$

5.4.4. F1 score

The F1 score ($\mathcal{S}_{\mathcal{F}1}$) provides an unbiased evaluation of the model's $\mathcal{R}_{c\ell}$ and $\mathcal{P}_{on}$ performance as follows

$$\mathcal{S}_{\mathcal{F}1} = 2 \times \frac{\mathcal{P}_{on} \times \mathcal{R}_{c\ell}}{\mathcal{P}_{on} + \mathcal{R}_{c\ell}} \tag{53}$$

Table 12 provides the relevant $\mathcal{S}_{\mathcal{F}1}, \mathcal{R}_{c\ell}$, and $\mathcal{P}_{on}$ for all the victim participants. The high values of evaluation metrics imply that, the effectiveness of the DT model in preserving data integrity and detecting attacks as it achieves an appropriate balance between $\mathcal{R}_{c\ell}$ and $\mathcal{P}_{on}$ and is good at correctly classifying occurrences.

5.5. Metrics for assessing the attack impact mitigation model

5.5.1. Mean absolute error

Mean absolute error, $\mathcal{E}_{\mathcal{MA}}$, is a commonly used loss function for regression-related DL-based models. In the context of DL, the performance of the constructed model is regularly evaluated by employing it as an evaluation metric. $\mathcal{E}_{\mathcal{MA}}$ measures the mean absolute error between the predicted and original values. Conceptually, $\mathcal{E}_{\mathcal{MA}}$ can be represented as follows if the dataset has $\mathcal{Y}$ samples, where $\hat{\mathcal{V}}_j$ and $\mathcal{V}_j$ stand for the predicted and original values, respectively, considering j^{th} as an independent parameter.

$$\mathcal{E}_{\mathcal{MA}} = \frac{1}{\mathcal{Y}} \sum_{j=1}^{\mathcal{Y}} |\hat{\mathcal{V}}_j - \mathcal{V}_j| \tag{54}$$

5.5.2. Root mean squared error

Root mean squared error ($\mathcal{E}_{\mathcal{RMS}}$) is another common loss function and assessment metric for DL-based regression issues. It determines the average squared deviation between the expected and actual values and takes the square root of that value. In the context of mathematics:

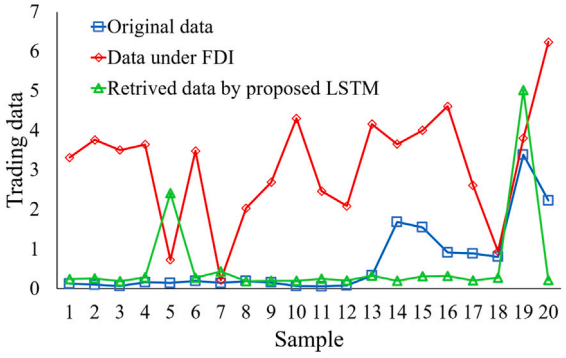
$$\mathcal{E}_{\mathcal{RMS}} = \sqrt{\frac{1}{\mathcal{Y}} \sum_{j=1}^{\mathcal{Y}} (\hat{\mathcal{V}}_j - \mathcal{V}_j)^2} \tag{55}$$

Table 13 lists the two evaluation metrics, $\mathcal{E}_{\mathcal{MA}}$ and $\mathcal{E}_{\mathcal{RMS}}$ for victim customers considering both the validation and test datasets. Lower

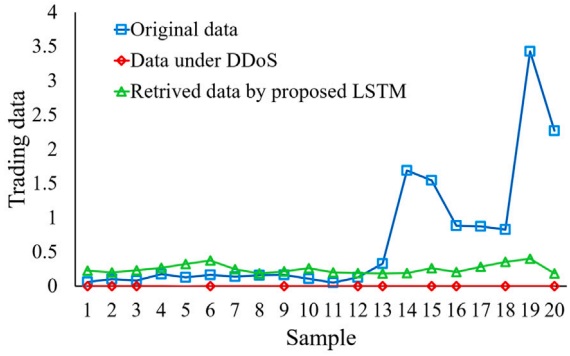
Table 13
Forecasting efficacy of the proposed LSTM model.

Identity of customer	$\mathcal{E}_{\mathcal{MA}}$ for validation set	$\mathcal{E}_{\mathcal{RMS}}$ for validation set	$\mathcal{E}_{\mathcal{MA}}$ for test set	$\mathcal{E}_{\mathcal{RMS}}$ for test set
$\mathcal{I}_5$	0.095	0.234	0.089	0.206
$\mathcal{I}_9$	0.062	0.102	0.064	0.1144
$\mathcal{I}_{11}$	0.091	0.197	0.082	0.163
$\mathcal{I}_{13}$	0.093	0.202	0.087	0.176
$\mathcal{I}_{22}$	0.058	0.093	0.061	0.116
$\mathcal{I}_{26}$	0.098	0.305	0.094	0.269

Here, $\mathcal{E}_{\mathcal{MA}}$ and $\mathcal{E}_{\mathcal{RMS}}$ stand for mean absolute error and root mean squared error, respectively.

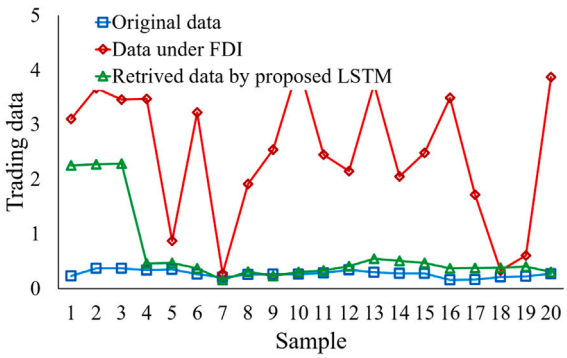


(a) Actual and predicted data for $\mathcal{I}_5$ considering FDI

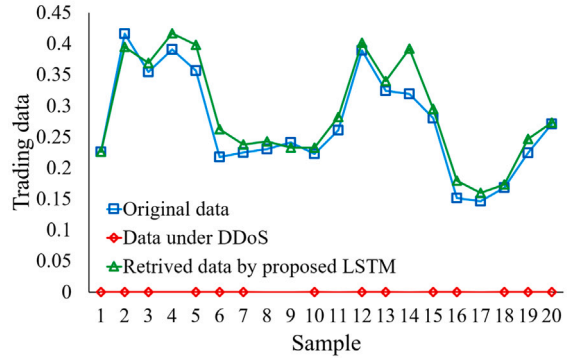


(b) Actual and predicted data for $\mathcal{I}_5$ considering DDoS

Fig. 8. Data predicting response of proposed LSTM for $\mathcal{I}_5$.



(a) Actual and predicted data for $\mathcal{I}_{13}$ considering FDI



(b) Actual and predicted data for $\mathcal{I}_{13}$ considering DDoS

Fig. 9. Data predicting response of proposed LSTM for $\mathcal{I}_{13}$.

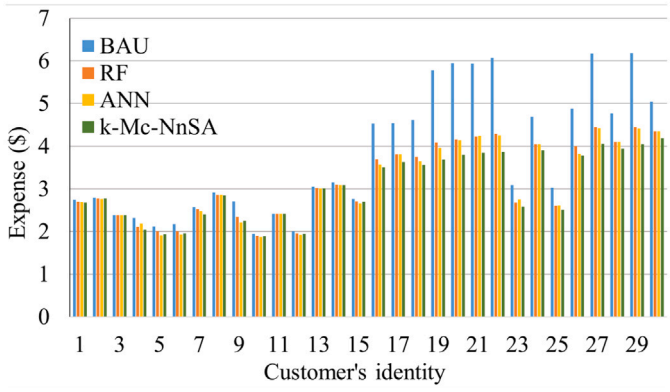
values of these metrics justify the efficacy of the LSTM-based proposed approach to retrieve the corrupted data.

The consistency in $\mathcal{E}_{\mathcal{MA}}$ and $\mathcal{E}_{\mathcal{RMS}}$ values for the validation and test sets, of all victim customers, justifies the model's robustness. Figs. 8 and 9 illustrate attack impact mitigation capability of the designed LSTM model, for two victim customers, $\mathcal{I}_5$ and $\mathcal{I}_{13}$ respectively as examples. Fig. 8a displays the original data retrieval ability of $\mathcal{I}_5$ during FDI attack, while 8b depicts its ability under DDoS attack. Fig. 9a and b do the same for victim participant $\mathcal{I}_{13}$. The visual representation depicts a small deviation from the original data at some points for both considered participants under FDI and DDoS attack variants. Because the 31st data point will be given as the output of the forecasting model for a given input data series ranging from 1st to 30th. The model will then use the 2nd through the 31st data points as input, with the 31st being the previously predicted data, if the 32nd data point is to be forecasted. As a result, for certain data points, the actual and anticipated data diverge slightly.

5.6. Effect of the optimal price model on LEM performance

This subsection analyzes the implications of the proposed k-Mc-NnSA based optimal trading price-giving model on P2P trading. In terms of sellers' earnings, $\mathcal{E}_{\mathcal{I}}$, and buyers' expenses, $\mathcal{E}_{\mathcal{B}}$, Fig. 10 compares the designed model to BAU, RF, and ANN to demonstrate its efficacy. Over a span of 24 hours, from 12 am to 11 : 55 pm, the previously established LEM structure is utilized to deploy the models. Fig. 10a demonstrates that, in comparison to BAU, RF, and ANN the suggested k-Mc-NnSA model yields better earnings for every participant operating as sellers. Zero earnings of particular customers in this figure indicate that they have no surplus energy for trading. Based on BAU, customers $\mathcal{I}_1, \mathcal{I}_5, \mathcal{I}_9$, and $\mathcal{I}_{12}$, for instance, make $ 3.4113, $ 3.2262, $ 2.777, and $ 3.5764 respectively for a sample day. For the same day, adopting RF gives those customers earnings of $ 3.5769, $ 3.4575, $ 2.8639, and $ 3.8438 respectively. For ANN these values become $ 3.8184, $ 3.7845, $ 3.0246, and $ 4.1684. When the proposed k-Mc-NnSA is used to provide $\mathcal{TP}_{\mathcal{O}}$, these

(a) Comparison of sellers' earning.



(b) Comparison of buyers' expense

Fig. 10. Comparison of financial benefits.

earnings rise to $ 3.9064, $ 3.8597, $ 3.0324, and $ 4.3213 accordingly for the day. The earnings show consistency across various customer identities, indicating stability and dependability in optimizing seller profits when the k-Mc-NnSA method is used.

Fig. 10b compares the $\mathcal{E}_{\mathcal{J}}$ with BAU, when RF, ANN and k-Mc-NnSA strategies are adopted for obtaining $\mathcal{TP}_{\mathcal{J}}$. The figure indicates that for each individual, k-Mc-NnSA results in a reduced $\mathcal{E}_{\mathcal{J}}$ when compared to BAU, RF, and ANN. For example, on a sample day, BAU indicates that the $\mathcal{E}_{\mathcal{J}}$ for clients $\mathcal{J}_7, \mathcal{J}_{13}, \mathcal{J}_{20}$, and $\mathcal{J}_{29}$, is $ 2.5751, $ 3.0428, $ 5.9382, and $ 6.1813 respectively. When adopting ANN these values become $ 2.4756, $ 3.012, $ 4.1423, and $ 4.4103. Using the proposed k-Mc-NnSA for obtaining $\mathcal{TP}_{\mathcal{J}}$ reduces $\mathcal{E}_{\mathcal{J}}$ values to $ 2.4057, $3.012, $ 3.7954, and $ 4.043 respectively on the same day. The lower expenses for every buyer with k-Mc-NnSA imply that it balances price optimization while providing affordability for buyers. As a result of the preceding explanation, it is clear that the k-Mc-NnSA maintains overall pricing balance and stability, making it a strong and efficient pricing strategy.

5.7. Computational overhead on edge devices

Most of the research dealing with the cyber security context of LEM is limited to theoretical aspects only; however, this work provides a comprehensive picture of the convergence and computing overhead of the suggested framework for real-world implementation. Proposed DT can identify data types instantly and the LSTM can replace the corrupted trading data immediately after detecting any compromised instance. When, these two models are integrated, i.e., 1st layer of proposed framework, approximately 0.5 s is required to identify and replace the corrupted data. The convergence findings in Table 14 show that the layered approach offers low latency without sacrificing functionality.

Table 14
Convergence of proposed framework.

Proposed models	Execution time
DT (Attack identification and categorization)	Instantly
LSTM (Attack impact mitigation)	Instantly
1st Layer (DT and LSTM)	0.5 s (a single data)
k-Mc-NnSA (Optimal trading price)	175 s, for the assumed market Structure over a 24-h period

5.8. Case study

How, the proposed secured bi-lateral energy transaction environment, first layer of the framework, facilitates maintaining the financial outcomes of seller and buyer in attack scenario is demonstrated in this section. Furthermore, the attack detection and mitigation layer's contribution to the establishment of a strong framework for safeguarding P2P energy trade is analyzed. Undoubtedly LSTM's data retrieval efficiency will play a vital role here. Figs. 11a–f illustrate the performance of attack impact mitigation by the designed LSTM model for victim customers $\mathcal{J}_5$, $\mathcal{J}_9$, $\mathcal{J}_{11}$, $\mathcal{J}_{13}$, $\mathcal{J}_{22}$, and $\mathcal{J}_{26}$ respectively for the test data set. From these figures it is clearly visible that, the predicted trading data values closely follow the actual values. However, some data points have a slight discrepancy between actual and predicted data, as seen in Fig. 11a–f.

In order to justify the proposed LSTM model's effectiveness in retrieving corrupted data from real-world contexts, its performance is evaluated by applying the dataset given by an Australian network service provider [55]. This dataset contains energy data of 300 customers over the period of 3 years with an interval of half an hour. For the training and testing purposes of our designed LSTM, the raw data of previously considered victim customers—$\mathcal{J}_5$, $\mathcal{J}_9$, $\mathcal{J}_{11}$, $\mathcal{J}_{13}$, $\mathcal{J}_{22}$, and $\mathcal{J}_{26}$—are used for a time horizon of 30 days. For the test set of real-world data, proposed LSTM model's performance in attack impact mitigation is exhibited by Fig. 12a–f for targeted customers $\mathcal{J}_5$, $\mathcal{J}_9$, $\mathcal{J}_{11}$, $\mathcal{J}_{13}$, $\mathcal{J}_{22}$, and $\mathcal{J}_{26}$ respectively.

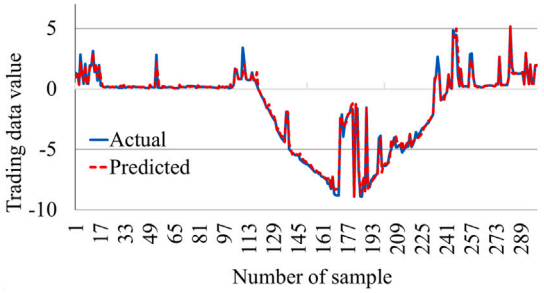
These Fig. 12a–f, show that the developed LSTM model can efficiently forecast trading data patterns using real-world data across a number of test instances. Although there are minor variations at abrupt transients, (e.g., sudden spikes in $\mathcal{J}_5$, $\mathcal{J}_9$, $\mathcal{J}_{11}$, and $\mathcal{J}_{22}$), the overall trajectory is well-preserved, indicating that the model may generalize to unseen data sufficiently.

For further investigation, our presumed LEM structure is used for a span of 24 h. Previously considered 6 customers—$\mathcal{J}_5, \mathcal{J}_9, \mathcal{J}_{11}, \mathcal{J}_{13}, \mathcal{J}_{22}$ and $\mathcal{J}_{26}$—are assumed to be the intruder's victims. 3 customers ($\mathcal{J}_5, \mathcal{J}_9$, and $\mathcal{J}_{11}$) have been designated under data integrity breaches, while the remaining 3($\mathcal{J}_{13}, \mathcal{J}_{22}$, and $\mathcal{J}_{26}$) are considered under data unavailability breaches. Under a security breach, it is estimated that 5 % of each prosumer's data has been compromised. To evaluate the effectiveness of the proposed scheme, two LEM-related essential terms, $\mathcal{P}_{\mathcal{J}}$ and $\mathcal{S}_{\mathcal{J}}$ in $, are addressed here. The deviation of the $\mathcal{P}_{\mathcal{J}}$, when RF, ANN, and k-Mc-NnSA are used, from the original profit during the compromised event can be described as follows, respectively:

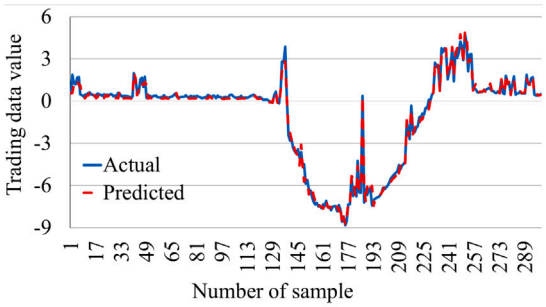
$$\hat{\mathcal{D}}_{\mathcal{J}\mathcal{O}\mathcal{C}\_\mathcal{RF}} = \frac{\mathcal{D}_{\mathcal{J}\mathcal{O}\_\mathcal{RF}} - \mathcal{D}_{\mathcal{J}\mathcal{C}\_\mathcal{RF}}}{\mathcal{D}_{\mathcal{J}\mathcal{O}\_\mathcal{RF}}} \times 100 \tag{56}$$

$$\hat{\mathcal{D}}_{\mathcal{J}\mathcal{O}\mathcal{C}\_\mathcal{ANN}} = \frac{\mathcal{D}_{\mathcal{J}\mathcal{O}\_\mathcal{ANN}} - \mathcal{D}_{\mathcal{J}\mathcal{C}\_\mathcal{ANN}}}{\mathcal{D}_{\mathcal{J}\mathcal{O}\_\mathcal{ANN}}} \times 100 \tag{57}$$

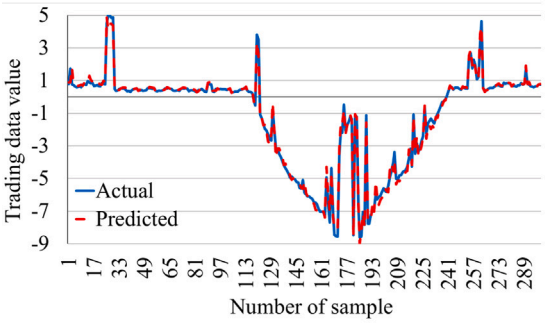
$$\hat{\mathcal{D}}_{\mathcal{J}\mathcal{O}\mathcal{C}\_\mathcal{P}} = \frac{\mathcal{D}_{\mathcal{J}\mathcal{O}\_\mathcal{P}} - \mathcal{D}_{\mathcal{J}\mathcal{C}\_\mathcal{P}}}{\mathcal{D}_{\mathcal{J}\mathcal{O}\_\mathcal{P}}} \times 100 \tag{58}$$



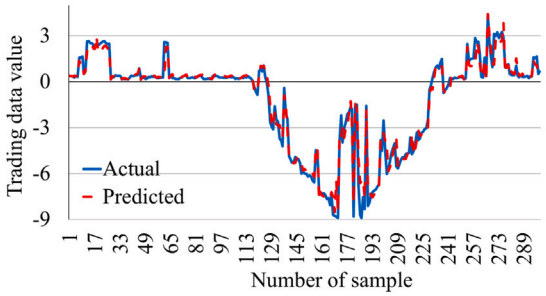
(a) LSTM's data retrieving capability of $\mathcal{J}_5$ for test set.



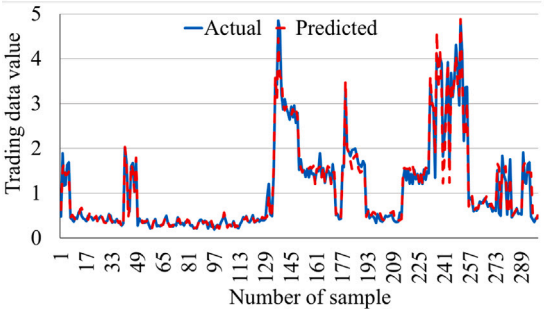
(b) LSTM's data retrieving capability of of $\mathcal{J}_9$ for test set.



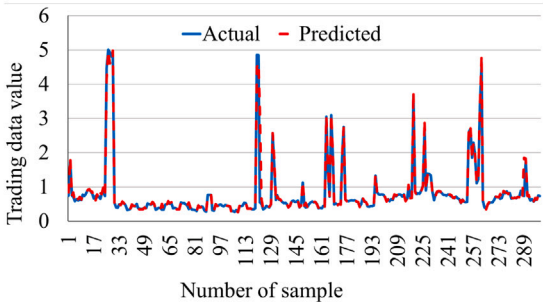
(c) LSTM's data retrieving capability of $\mathcal{J}_{11}$ for test set.



(d) LSTM's data retrieving capability of of $\mathcal{J}_{13}$ for test set.



(e) LSTM's data retrieving capability of $\mathcal{J}_{22}$ for test set.



(f) LSTM's data retrieving capability of of $\mathcal{J}_{26}$ for test set.

Fig. 11. Data retrieving performance of LSTM for test set.

here $\mathcal{D}_{\mathcal{J}o\_\mathcal{RF}}, \mathcal{D}_{\mathcal{J}o\_\mathcal{ANN}}, \mathcal{D}_{\mathcal{J}o\_\mathcal{P}}$ represent the $\mathcal{P}_{\mathcal{J}}$ when RF, ANN and k-Mc-NnSA models are used for providing $\mathcal{TP}_{\mathcal{O}}$ respectively under no attack conditions, while $\mathcal{D}_{\mathcal{J}o\_\mathcal{RF}}, \mathcal{D}_{\mathcal{J}o\_\mathcal{ANN}} \mathcal{D}_{\mathcal{J}o\_\mathcal{P}}$ represent the same under compromised conditions. The percentage deviation between $\mathcal{P}_{\mathcal{J}}$ under normal and corrupted conditions for RF, ANN and k-Mc-NnSA is represented by $\hat{\mathcal{D}}_{\mathcal{J}oc\_\mathcal{RF}}, \hat{\mathcal{D}}_{\mathcal{J}oc\_\mathcal{ANN}}$, and $\hat{\mathcal{D}}_{\mathcal{J}oc\_\mathcal{P}}$, respectively. Maintaining above pattern, $\hat{\mathcal{D}}_{\mathcal{B}oc\_\mathcal{RF}}, \hat{\mathcal{D}}_{\mathcal{B}oc\_\mathcal{ANN}}$ and $\hat{\mathcal{D}}_{\mathcal{B}oc\_\mathcal{P}}$ are calculated for $\mathcal{S}_{\mathcal{B}}$ for RF, ANN and k-Mc-NnSA, respectively as follows:

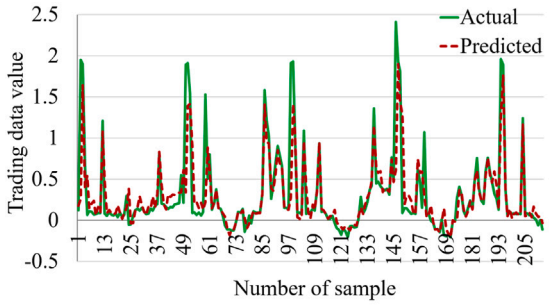
$$\hat{\mathcal{D}}_{\mathcal{B}oc\_\mathcal{RF}} = \frac{\mathcal{D}_{\mathcal{B}o\_\mathcal{RF}} - \mathcal{D}_{\mathcal{B}c\_\mathcal{RF}}}{\mathcal{D}_{\mathcal{B}o\_\mathcal{RF}}} \times 100 \tag{59}$$

$$\hat{\mathcal{D}}_{\mathcal{B}oc\_\mathcal{ANN}} = \frac{\mathcal{D}_{\mathcal{B}o\_\mathcal{ANN}} - \mathcal{D}_{\mathcal{B}c\_\mathcal{ANN}}}{\mathcal{D}_{\mathcal{B}o\_\mathcal{ANN}}} \times 100 \tag{60}$$

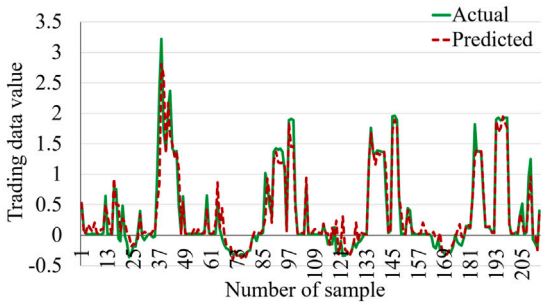
$$\hat{\mathcal{D}}_{\mathcal{B}oc\_\mathcal{P}} = \frac{\mathcal{D}_{\mathcal{B}o\_\mathcal{P}} - \mathcal{D}_{\mathcal{B}c\_\mathcal{P}}}{\mathcal{D}_{\mathcal{B}o\_\mathcal{P}}} \times 100 \tag{61}$$

Under cyberattack conditions, Table 15 presents the $\mathcal{S}_{\mathcal{B}}$ and $\mathcal{P}_{\mathcal{J}}$ deviation from standard operating conditions when RF, ANN, and k-Mc-NnSA are used for suggesting trading price. The table's numerical analysis demonstrates that, when employing k-Mc-NnSA for $\mathcal{TP}_{\mathcal{O}}$, the deviation in seller and buyer financial profit is slightly more than RF for some participants, as example—sellers: $\mathcal{J}_5, \mathcal{J}_6, \mathcal{J}_8, \mathcal{J}_{10}, \mathcal{J}_{12}, \mathcal{J}_{15}$; buyers: $\mathcal{J}_{18}, \mathcal{J}_{19}, \mathcal{J}_{24}, \mathcal{J}_{30}$—but the average deviation is lower when all of the participants are taken into consideration. The average value of $\hat{\mathcal{D}}_{\mathcal{J}oc\_\mathcal{P}}$ and $\hat{\mathcal{D}}_{\mathcal{B}oc\_\mathcal{P}}$ are 1.279 % and 0.388 % respectively which is less than the respective values when RF—1.447 % and 0.427 %—and ANN—1.288 % and 0.476 %—models are used for trading price giving purpose. Hence, obtaining $\mathcal{TP}_{\mathcal{O}}$ through proposed k-Mc-NnSA not only facilitates $\mathcal{P}_{\mathcal{J}}$ and $\mathcal{S}_{\mathcal{B}}$, as depicted by Fig. 6, but also reduces their deviation in the event of cyberattack when compared to RF and ANN.

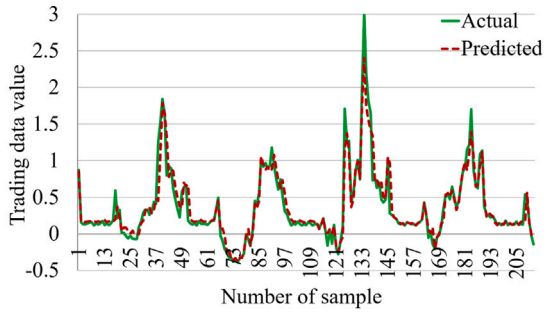
The variation in financial benefits for both sellers and buyers is investigated in this portion in order to demonstrate the usefulness of the proposed framework in sustaining market efficiency under cyber attack conditions. Deviations in $\mathcal{S}_{\mathcal{B}}$ and $\mathcal{P}_{\mathcal{J}}$ from their original values will now be investigated with respect to the proposed secured P2P trading environment. Table 16 enlists the numeric values of $\hat{\mathcal{D}}_{\mathcal{J}oc\_\mathcal{P}}, \hat{\mathcal{D}}_{\mathcal{J}of\_\mathcal{P}}, \hat{\mathcal{D}}_{\mathcal{B}oc\_\mathcal{P}}$ and $\hat{\mathcal{D}}_{\mathcal{B}of\_\mathcal{P}}$ which represent the deviation in $\mathcal{P}_{\mathcal{J}}$ and $\mathcal{S}_{\mathcal{B}}$ from normal trading conditions under corrupted and proposed frameworks, respectively. Mathematically, $\hat{\mathcal{D}}_{\mathcal{J}of\_\mathcal{P}}$ and $\hat{\mathcal{D}}_{\mathcal{B}of\_\mathcal{P}}$ can be represented as



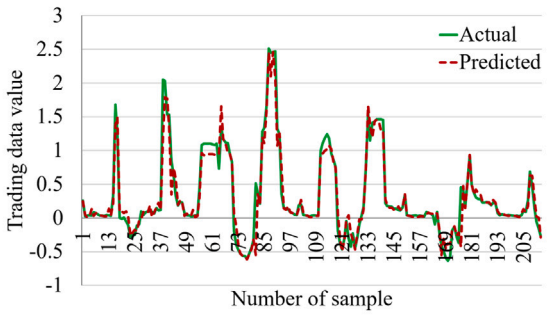
(a) LSTM's data retrieving of $\mathcal{J}_5$ for test set.



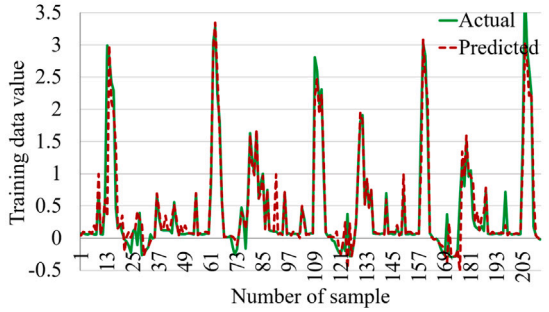
(b) LSTM's data retrieving of of $\mathcal{J}_9$ for test set.



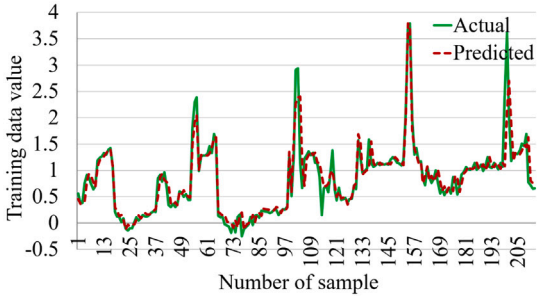
(c) LSTM's data retrieving of $\mathcal{J}_{11}$ for test set.



(d) LSTM's data retrieving of of $\mathcal{J}_{13}$ for test set.



(e) LSTM's data retrieving of $\mathcal{J}_{22}$ for test set.



(f) LSTM's data retrieving of of $\mathcal{J}_{26}$ for test set.

Fig. 12. Data retrieving performance of LSTM for test set considering real-world data.

follows:

$$\hat{\mathcal{D}}_{\mathcal{J}of\_\mathcal{P}} = \frac{\mathcal{D}_{\mathcal{J}o\_\mathcal{P}} - \mathcal{D}_{\mathcal{J}f\_\mathcal{P}}}{\mathcal{D}_{\mathcal{J}o\_\mathcal{P}}} \times 100 \tag{62}$$

and

$$\hat{\mathcal{D}}_{\mathcal{B}of\_\mathcal{P}} = \frac{\mathcal{D}_{\mathcal{B}o\_\mathcal{P}} - \mathcal{D}_{\mathcal{B}f\_\mathcal{P}}}{\mathcal{D}_{\mathcal{B}o\_\mathcal{P}}} \times 100 \tag{63}$$

During compromised operating conditions, the average values of $\hat{\mathcal{D}}_{\mathcal{J}oc\_\mathcal{P}}$ and $\hat{\mathcal{D}}_{\mathcal{B}oc\_\mathcal{P}}$ are 1.279 % and 0.388 % respectively. This suggests that cyberthreats can have a major detrimental influence on financial efficiency; for both seller and buyer; and fairness of market. However, these variations are significantly decreased to 0.008 % for sellers and 0.0147 % for buyers under the designed secured P2P environment. These negligible variations in $\mathcal{P}_{\mathcal{J}}$ and $\mathcal{S}_{\mathcal{B}}$ reveal that, they closely match the values obtained in typical (attack-free) circumstances, demonstrating that the designed framework effectively restores market efficiency and financial consistency. Hence, our proposed trading security layer adds an effective dimension to the financial benefiting layer and makes the total framework a robust one against cyberthreats.

To analyze how an attacker can profit financially from the deployed cyberattack plan, let us assume that the attacker is one of the victim customers. Table 17 represents the changing market dynamics after deploying the attack strategy mentioned in Subsection 3.3 in a summarized manner. It is noticeable that, when generation through solar PV exists, the injection of FDI with positive bias factor creates artificial surplus energy, which can be sold in the market, allowing the attacker to gain as a seller (e.g., $\mathcal{J}_5$, $\mathcal{J}_9$, $\mathcal{J}_{11}$, and $\mathcal{J}_{13}$). In the absence of solar PV energy, FDI deployment fails to increase seller profit; instead, it generates false demand that compels a customer to act as a buyer, here customers $\mathcal{J}_{22}$, and $\mathcal{J}_{26}$. In the event of a DDoS attack, the victim who has PV generation ($\mathcal{J}_{13}$) is forced into a buying role because the blocked data stops surplus energy from being transferred. In contrast, the DDoS attack only affects demand–supply matching and forces the market participants ($\mathcal{J}_{22}$, and $\mathcal{J}_{26}$) to operate as self-sufficient when there is no PV generation.

Under this situation, Tables 17 and 19indicate the variation in seller, $\hat{\mathcal{D}}_{\mathcal{J}oc\_\mathcal{P}}$; obtained from (58); and buyer, $\hat{\mathcal{D}}_{\mathcal{B}oc\_\mathcal{P}}$; following (61); benefits (in %) respectively, considering previously mentioned attack strategy and market structure for a span of 24 hours, during the attack compared to the normal circumstances. Thus, a positive value during an attack scenario represents the attacker's financial benefit.

Tables 18 and 19 clearly show that $\mathcal{J}_5$, $\mathcal{J}_9$, and $\mathcal{J}_{11}$ make money as sellers, $\mathcal{J}_{22}$ and $\mathcal{J}_{26}$ earn money as buyers, while $\mathcal{J}_{13}$ receives money as

Table 15
Deviation in seller and buyer financial benefits under cyberattack for RF, ANN and k-Mc-NnSA.

Identity of prosumer	$\hat{\mathcal{D}}_{soc\_\mathcal{RF}}$ (%)	$\hat{\mathcal{D}}_{soc\_ANN}$ (%)	$\hat{\mathcal{D}}_{soc\_\mathcal{P}}$ (%)	$\hat{\mathcal{D}}_{boc\_\mathcal{RF}}$ (%)	$\hat{\mathcal{D}}_{boc\_ANN}$ (%)	$\hat{\mathcal{D}}_{boc\_\mathcal{P}}$ (%)
$\mathcal{I}_1$	3.321	2.678	2.484	0	0	0
$\mathcal{I}_2$	2.551	3.376	1.908	0	0	0
$\mathcal{I}_3$	4.843	2.828	2.837	0	0	0
$\mathcal{I}_4$	6.749	6.098	6.093	0	0	0
$\mathcal{I}_5$	0.519	0.304	1.247	0	0	0
$\mathcal{I}_6$	3.694	4.171	4.251	0	0	0
$\mathcal{I}_7$	1.455	0.699	0.951	0	0	0
$\mathcal{I}_8$	2.620	4.089	3.034	0	0	0
$\mathcal{I}_9$	1.611	0	1.488	0	0	0
$\mathcal{I}_{10}$	1.959	3.191	2.326	0	0	0
$\mathcal{I}_{11}$	9.274	5.791	6.017	1	0	0
$\mathcal{I}_{12}$	2.731	3.379	4.994	0	0	0
$\mathcal{I}_{13}$	0.520	0.239	0.133	11.808	14.286	11.632
$\mathcal{I}_{14}$	1.331	1.025	0.325	0	0	0
$\mathcal{I}_{15}$	0.230	0.784	0.286	0	0	0
$\mathcal{I}_{16}$	0	0	0	0.005	0	0
$\mathcal{I}_{17}$	0	0	0	0.056	0	0.051
$\mathcal{I}_{18}$	0	0	0	0.224	0	0.484
$\mathcal{I}_{19}$	0	0	0	0.019	0	0.033
$\mathcal{I}_{20}$	0	0	0	0.008	0	0.065
$\mathcal{I}_{21}$	0	0	0	0.033	0	0.114
$\mathcal{I}_{22}$	0	0	0	12.393	14.149	12.443
$\mathcal{I}_{23}$	0	0	0	0.318	0.517	0.328
$\mathcal{I}_{24}$	0	0	0	0.071	0	0.163
$\mathcal{I}_{25}$	0	0	0	0	0	0.022
$\mathcal{I}_{26}$	0	0	0	20.101	22.916	20.059
$\mathcal{I}_{27}$	0	0	0	0.003	0	0.153
$\mathcal{I}_{28}$	0	0	0	0.028	0	0
$\mathcal{I}_{29}$	0	0	0	0.077	0	0.049
$\mathcal{I}_{30}$	0	0	0	0.020	0	0.027
Average	**1.447**	**1.288**	**1.279**	**0.427**	**0.476**	**0.388**

Table 16
Deviation in seller and buyer financial benefits under cyberattack and proposed secured conditions for k-Mc-NnSA.

Identity of prosumer	$\hat{\mathcal{D}}_{soc\_\mathcal{P}}$(%)	$\hat{\mathcal{D}}_{sof\_\mathcal{P}}$(%)	$\hat{\mathcal{D}}_{boc\_\mathcal{P}}$(%)	$\hat{\mathcal{D}}_{bof\_\mathcal{P}}$(%)
$\mathcal{I}_1$	2.484	0.04	0	0
$\mathcal{I}_2$	1.908	0	0	0
$\mathcal{I}_3$	2.837	0	0	0
$\mathcal{I}_4$	6.092	0.031	0	0
$\mathcal{I}_5$	1.246	0.031	0	0
$\mathcal{I}_6$	4.250	0.02	0	0
$\mathcal{I}_7$	0.951	0	0	0
$\mathcal{I}_8$	3.034	0	0	0.278
$\mathcal{I}_9$	1.487	0.04	0	0.011
$\mathcal{I}_{10}$	2.325	0	0	0
$\mathcal{I}_{11}$	6.017	0.02	0	0
$\mathcal{I}_{12}$	4.993	0.027	0	0
$\mathcal{I}_{13}$	0.133	0	11.632	0
$\mathcal{I}_{14}$	0.324	0.027	0	0
$\mathcal{I}_{15}$	0.286	0	0	0
$\mathcal{I}_{16}$	0	0	0	0
$\mathcal{I}_{17}$	0	0	0.051	0
$\mathcal{I}_{18}$	0	0	0.483	0
$\mathcal{I}_{19}$	0	0	0.032	0.008
$\mathcal{I}_{20}$	0	0	0.065	0
$\mathcal{I}_{21}$	0	0	0.113	0
$\mathcal{I}_{22}$	0	0	12.442	0.042
$\mathcal{I}_{23}$	0	0	0.328	0.037
$\mathcal{I}_{24}$	0	0	0.162	0.007
$\mathcal{I}_{25}$	0	0	0.022	0
$\mathcal{I}_{26}$	0	0	20.059	0.055
$\mathcal{I}_{27}$	0	0	0.152	0
$\mathcal{I}_{28}$	0	0	0	0
$\mathcal{I}_{29}$	0	0	0.049	0
$\mathcal{I}_{30}$	0	0	0.027	0
Average	**1.279**	**0.008**	**0.388**	**0.0147**

Table 17
Effect of attack on market dynamics.

Attack types	Solar PV	Consumer	Manipulation effect	Customer's role	
				As seller	As buyer
FDI	✓	$\mathcal{I}_5$, $\mathcal{I}_9$, $\mathcal{I}_{11}$, $\mathcal{I}_{13}$	Creating artificial surplus	✓	×
FDI	×	$\mathcal{I}_{22}$, $\mathcal{I}_{26}$	Creating artificial demand	×	✓
DDoS	✓	$\mathcal{I}_{13}$	Forced into buyer role	×	✓
DDoS	×	$\mathcal{I}_{22}$, $\mathcal{I}_{26}$	Forced into self-sufficient	×	×

Table 18
Financial benefit of attacker as seller.

Identity of customer	$\hat{\mathcal{D}}_{soc\_\mathcal{P}}$ (%)	Interpretation
$\mathcal{I}_5$	1.247	Attacker gains money
$\mathcal{I}_9$	1.488	Attacker gains money
$\mathcal{I}_{11}$	6.017	Attacker gains money
$\mathcal{I}_{13}$	0.133	Attacker gains money
$\mathcal{I}_{22}$	0.000	No gain
$\mathcal{I}_{26}$	0.000	No gain

Table 19
Financial benefit of attacker as buyer.

Identity of customer	$\hat{\mathcal{D}}_{boc\_\mathcal{P}}$ **(%)**	Interpretation
$\mathcal{I}_5$	0.000	No gain
$\mathcal{I}_9$	0.000	No gain
$\mathcal{I}_{11}$	0.000	No gain
$\mathcal{I}_{13}$	11.632	Attacker gains large money
$\mathcal{I}_{22}$	12.442	Attacker gains major money
$\mathcal{I}_{26}$	20.059	Attacker gains huge money

both seller and buyer. This demonstrates that when attackers are among the target consumers, they may profit from the attack to gain an unfair advantage in the market.

6. Conclusion

IoT-based energy trading platforms, such as LEMs, enable consumers to actively engage in energy generation, consumption, and trading as energy systems transition to decentralized paradigms. Nevertheless, this digital empowerment carries risks of cyberthreats, endangering participants' trust in these networks as well as their financial gains. Therefore, protecting IoT-based energy trading is essential for maintaining market integrity as well as safeguarding prosumers' investments and interests, as these systems are crucial for sustainable energy practices. To meet this increasing demand, this research proposes an extensive structure based on AI to ensure the safety and dependability of energy trading in LEMs. Utilizing AI, the proposed framework helps identify and mitigate the effects of cyberthreats on bilateral energy trade while also providing an optimal trading price-setting model to improve profitability for both sellers and buyers.

Through the first layer, the framework preserves the availability and integrity of trading data, enabling smooth and safe energy transactions between market participants. Even in compromised situations, the framework guarantees a stable trading environment by identifying and substituting corrupted data. The framework also preserves the fairness of the LEM by guaranteeing the financial gain of both buyers and sellers by providing an optimal trading price. Both the numerical and graphical findings show that the suggested framework secures energy trade, encourages fair market behaviors, and successfully lessens the effects of cyberattacks.

The accuracy of the proposed DT scheme in identifying attacks for each of the victim prosumers—$\mathcal{I}_5, \mathcal{I}_9, \mathcal{I}_{11}, \mathcal{I}_{13}, \mathcal{I}_{22}$ and $\mathcal{I}_{26}$—has been 98.51 %, 97.77 %, 97.77 %, 99.26 %, 98.76 %, and 99.50 % respectively. The efficiency of the designed LSTM for recovering original trading data is supported by the values of $\mathcal{E}_{\mathcal{RMS}}$—0.206, 0.1144, 0.163, 0.176, 0.116, 0.269 and $\mathcal{E}_{\mathcal{MA}}$—0.089, 0.064, 0.082, 0.087, 0.061, 0.094, respectively for the victim participants. Additionally, the comparison of customers' financial benefits with BAU, RF, and ANN supports the effectiveness of the proposed k-Mc-NnSA in providing optimal trading prices. The small values of $\hat{\mathcal{D}}_{\mathcal{I}of\_\mathcal{P}}$ and $\hat{\mathcal{D}}_{\mathcal{B}of\_\mathcal{P}}$—0.008 % and 0.0147 %—reveal how effectively the proposed security layer helps to maintain financial benefits among market participants during cyberattack. As the energy industry develops further, the findings of this work can be crucial to ensure secure and efficient energy trading, considering the cyber-dependent participation of multiple entities. This work does not assess how a cyberattack affects the financial gain of individual participants. Future research will concentrate on a thorough examination of how cyberattacks affect LEM trading financially. We also intend to expand our future research by integrating sophisticated attack modeling that guarantees financial gain at every attack instant in order to assess the efficacy of our suggested framework. Algorithmic enhancements, such as a customized loss function for joint security-pricing optimization, could be a potential direction for future work. Additionally, it will explore the integration of deep reinforcement learning-based pricing models to enable comprehensive comparisons with advanced optimization techniques and enhance the robustness of the proposed framework.

CRediT authorship contribution statement

Fariya Tabassum: Writing – original draft, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **M. Imran Azim:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Investigation, Conceptualization. **Md. Rashidul Islam:** Writing – review & editing, Visualization, Validation, Conceptualization. **M.A. Rahman:** Writing – original draft, Software, Methodology, Formal analysis. **Liaqat Ali:** Writing – review & editing, Visualization, Supervision. **Md. Mahfuzur Rahman:** Writing – review & editing, Visualization, Supervision. **M.J. Hossain:** Writing – review & editing, Visualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

[1] You Z, Lumpp SD, Doepfert M, Tzscheutschler P, Goebel C. Leveraging flexibility of residential heat pumps through local energy markets. Appl Energy 2024;355(122269).

[2] Ali L, Azim MI, Peters J, Pashajavid E. Expediting battery investment returns for residential customers utilising spot price-aware local energy exchanges. Energy 2024 Oct.;306(132458):1–12.

[3] Azim MI, Alam MR, Tushar W, Saha TK, Yuen C. A cooperative P2P trading framework: developed and validated through hardware-in-loop. IEEE Trans Smart Grid 2023 Jul.;14(4):2999–3015.

[4] Gourisetti SNG, Sebastian-Cardenas DJ, Bhattarai B, Wang P, Widergren S, Borkum M, et al. Blockchain smart contract reference framework and program logic architecture for transactive energy systems. Appl Energy 2021;304(117860).

[5] Zheng Y, Xue X, Xi S, Wang X. Balancing possibilist-probabilistic risk assessment for smart energy hubs: enabling secure peer-to-peer energy sharing with ccus technology and cyber-security. Energy 2024;304(132102).

[6] Erdayandi K, Mustafa MA. PP-LEM: efficient and privacy-preserving clearance mechanism for local energy markets. Sustain Energy Grids Netw 2024; 39(101477).

[7] Ali L, Azim MI, Ojha NB, Peters J, Bhandari V, Menon A, et al. Integrating forecasting service and Gen2 blockchain into a local energy trading platform to promote sustainability goals. IEEE Access 2023 Dec.;12:2941–64.

[8] Ali L, Azim MI, Peters J, Ojha NB, Simoes M, Muyeen S. Integrating Gen3 blockchain into a transactive energy market for ders orchestration. IEEE Trans Ind Appl 2025 May-Jun;61(3):5103–15.

[9] Jin AS, Sanders KT. Characterizing residential sector load curves from smart meter datasets. Appl Energy 2024;366(123316).

[10] Munoz O, Ruelas A, Rosales-Escobedo PF, Acuña A, Suastegui A, Lara F, et al. Development of an IoT smart energy meter with power quality features for a smart grid architecture. Sustain Comput Inform Syst 2024;43(100990).

[11] Azim MI, Tushar W, Saha TK, Yuen C, Smith D. Peer-to-peer kilowatt and negawatt trading: a review of challenges and recent advances in distribution networks. Renew Sustain Energy Rev 2022;169(112908).

[12] Hseiki HA, El-Hajj AM, Ajra YO, Hija FA, Haidar AM. A secure and resilient smart energy meter. IEEE Access 2024;12:3114–25.

[13] Mitra S, Chakraborty B, Mitra P. Smart meter data analytics applications for secure, reliable and robust grid system: survey and future directions. Energy 2024;289(129920).

[14] Rouhani SH, Su C-L, Mobayen S, Razmjooy N, Elsisi M. Cyber resilience in renewable microgrids: a review of standards, challenges, and solutions. Energy 2024;289(133081).

[15] Parvez I, Aghili M, Riggs H, Sundararajan A, Sarwat AI, Srivastava AK. A novel authentication management for the data security of smart grid. IEEE Open Access J Power Energy 2024;11(218).

[16] Kabir F, Araghi TK, Megías D. Privacy-preserving protocol for high-frequency smart meters using reversible watermarking and paillier encryption. Comput Electr Eng 2024;119(109497).

[17] Sakr HA, Fouda MM, Ashour AF, Abdelhafeez A, El-Afifi MI, Abdellah MR. Machine learning-based detection of DDoS attacks on IoT devices in multi-energy systems. Egypt Inform J 2024;28(100540).

[18] Ayub MF, Li X, Mahmood K, Shamshad S, Saleem MA, Omar M. Secure consumer-centric demand response management in resilient smart grid as industry 5.0 application with blockchain-based authentication. IEEE Trans Consumer Electron 2023;70:1370–9.

[19] Hizal S, Cavusoglu U, Akgun D. A novel deep learning-based intrusion detection system for IoT DDoS security. Internet Things 2024;28(101336).

[20] Diaba SY, Elmusrati M. Proposed algorithm for smart grid DDoS detection based on deep learning. Neural Netw 2023;159:175–84.

[21] Hayat RF, Aurangzeb S, Aleem M, Srivastava G, Lin JC-W. ML-DDoS: a blockchain-based multilevel DDoS mitigation mechanism for IoT environments. IEEE Trans Eng Manag 2022;71:12605–18.

[22] Lei W, Pang Z, Wen H, Hou W, Han W. FDI attack detection at the edge of smart grids based on classification of predicted residuals. IEEE Trans Ind Informatics 2022;18(12):9302–11.

[23] Soto EA, Bosman LB, Wollega E, Leon-Salas WD. Peer-to-peer energy trading: a review of the literature. Appl Energy 2021;283(116268).

[24] Tushar W, Nizami S, Azim MI, Yuen C, Smith DB, Saha T, et al. Peer-to-peer energy sharing: a comprehensive review. Found. Trends Electr. Energy Syst. 2023 Feb.;6(1):1–82.
[25] Hussain S, Azim MI, Lai C, Eicker U. Smart home integration and distribution network optimization through transactive energy framework - a review. Appl Energy 2025 Oct.;395(126193):1–23.
[26] Jia H, Wang X, Jin X, Cheng L, Mu Y, Yu X, et al. Optimal pricing of integrated community energy system for building prosumers with P2P multi-energy trading. Appl Energy 2024;365(123259).
[27] Amin W, Ahmad F, Umer K, Khawaja AH, Afzal M, Ahmad SA, et al. An effective pricing mechanism for electricity trading considering customer preference and reserved price in direct P2P electricity market under uncertainty in grid supply. IEEE Access 2022;10:96197–211.
[28] Meng Y, Ma G, Ye Y, Yao Y, Li W, Li T. Design of P2P trading mechanism for multi-energy prosumers based on generalized nash bargaining in GCT-CET market. Appl Energy 2024;371:123640:1–12.
[29] Izanlo A, Sheikholeslami A, Gholamian SA, Kazemi MV, Hosseini SN. A combination of MILP and game theory methods for P2P energy trading by considering network constraints. Appl Energy 2024;374:123916:1–12391612.
[30] Zhou Y, Lund PD. Peer-to-peer energy sharing and trading of renewable energy in smart communities- trading pricing models, decision-making and agent-based collaboration. Renew Energy 2023;207:177–93.
[31] Wang N, Li J, Ho SS, Qiu C. Distributed machine learning for energy trading in electric distribution system of the future. Electr J 2021;34:106883.
[32] Qayyum F, Jamil H, Jamil F, Kim D. Predictive optimization based energy cost minimization and energy sharing mechanism for peer-to-peer nanogrid network. IEEE Access 2022;10:23593–604.
[33] Wang J, Li L, Zhang J. Deep reinforcement learning for energy trading and load scheduling in residential peer-to-peer energy trading market. Int J Electr Power Energy Syst 2023;147:108885.
[34] Tang Q, Guo H, Zheng K, Chen Q. Forecasting individual bids in real electricity markets through machine learning framework. Appl Energy 2024;363:123053.
[35] Srivastava M, Tiwari PK. A profit driven optimal scheduling of virtual power plants for peak load demand in competitive electricity markets with machine learning based forecasted generations. Energy 2024;310:133077.
[36] Bampos ZN, Laitsos VM, Afentoulis KD, Vagropoulos SI, Biskas PN. Electric vehicles load forecasting for day-ahead market participation using machine and deep learning methods. Appl Energy 2024;360:122801.
[37] Singh K, Singha N. Credit-based energy trading system using blockchain and machine learning. J Supercomput 2024;80(11):15386–407.
[38] Tabassum F, Islam MR, Azim MI, Rahman MA, Faruque MO, Shezan SKA, et al. Secured energy data transaction for prosumers under diverse cyberattack scenarios. Sustain Energy Grids Netw 2024;40(101555).
[39] Melendez KA, Matamala Y. Adversarial attacks in demand-side electricity markets. Appl Energy 2025;377:124615.
[40] Mohammadi S, Eliassen F, Zhang Y, Jacobsen H-A. Detecting false data injection attacks in peer to peer energy trading using machine learning. IEEE Trans Dependable Secure Comput 2021;19(5):3417–31.
[41] Cui M, Wang J, Yue M. Machine learning-based anomaly detection for load forecasting under cyberattacks. IEEE Trans Smart Grid 2019;10(5):5724–34.
[42] Rahman MA, Islam MR, Md. Alamgir Hossain MSR, Hossain MJ, Gray EMA. Resiliency of forecasting methods in different application areas of smart grids: a review and future prospects. Eng Appl Artif Intell 2024;135:108785.
[43] Zografopoulos I, Hatziargyriou ND, Konstantinou C. Distributed energy resources cybersecurity outlook: vulnerabilities, attacks, impacts, and mitigations. IEEE Syst J 2023;17(4):6695–709.
[44] Kadri MR, Abdelli A, Othman JB, Mokdad L. Survey and classification of Dos and DDos attack detection and validation approaches for IoT environments. Internet Of Things 2024;25:101021.
[45] Wang H, Genghui L, Wang Z. Fast SVM classifier for large-scale classification problems. Inf Sci 2023;642:119136.
[46] Delilbasic Amer, Saux L ,et al. A single-step multiclass SVM based on quantum annealing for remote sensing data classification. IEEE J Sel Top Appl Earth Obs Remote Sens 2023;17:1434–45.
[47] Krishna TB, Kokil P. Automated classification of common maternal fetal ultrasound planes using multi-layer perceptron with deep feature integration. Biomed Signal Process Control 2023;86:105283.
[48] Kumar V, Agrawal S. A multi-layer perceptron–Markov chain based LULC change analysis and prediction using remote sensing data in prayagraj district, India. Environ Monit Assess 2023;195(5):619.
[49] Nieto PJG, García–Gonzalo E, María Paredes–Sánchez B, Pablo Paredes–Sánchez J. Modelling energy performance of residential dwellings by using the mars technique, SVM-based approach, MLP neural network and M5 model tree. Appl Energy 2023;341:121074.
[50] Zehuan H, Gao Y, Siyu J, Mae M, Imaizumi T. Improved multistep ahead photovoltaic power prediction model based on LSTM and self-attention with weather forecast data. Appl Energy 2024;359:122709.
[51] Sun Z, Wang G, Pengfei L, Wang H, Zhang M, Liang X. An improved random forest based on the classification accuracy and correlation measurement of decision trees. Expert Syst Appl 2024;237:121549.
[52] Tian Y, Yue Z, Zhang R, Zhao X, Zheng B, Zhou X. Approximate nearest neighbor search in high dimensional vector databases: current research and future directions. IEEE Data Eng Bull 2023;(3):39–54.
[53] Ikotun AM, Ezugwu AE, Abualigah L, Abuhaija B, Heming J. k-means clustering algorithms: a comprehensive review, variants analysis, and advances in the era of big data. Inf Sci 2023;622:178–210.
[54] Haize H, Liu J, Zhang X, Fang M, "An effective and adaptable k-means algorithm for big data cluster analysis". Pattern Recognit 2023;139:109404.
[55] Solar home electricity data," https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data [Accessed on May 15, 2024].