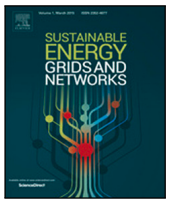
Sustainable Energy, Grids and Networks 40 (2024) 101555

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

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



Secured energy data transaction for prosumers under diverse cyberattack scenarios

Fariya Tabassum [a], Md. Rashidul Islam [b], M. Imran Azim [c,*], M.A. Rahman [b], Md. Omer Faruque [b], Sk.A. Shezan [d], M.J. Hossain [e]

[a] Department of Electrical & Computer Engineering, Rajshahi University of Engineering & Technology, Rajshahi 6204, Bangladesh
[b] Department of Electrical & Electronic Engineering, Rajshahi University of Engineering & Technology, Rajshahi 6204, Bangladesh
[c] Department of Electrical and Computer Systems Engineering, Monash University, Clayton, VIC 3068, Australia
[d] Department of Electrical Engineering, Engineering Institute of Technology, Melbourne Campus, VIC 3001, Australia
[e] School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia

ARTICLE INFO

Keywords:
Secured energy trading
Prosumers
Artificial intelligence
Internet of things
Outlier data

ABSTRACT

Due to the increasing use of renewable energy sources and the advancement of smart grid technology, bilateral energy transactions between prosumers have attracted significant interest as a potential solution for efficient and decentralized energy distribution. Prosumers can establish direct energy exchanges by utilizing internet of things (IoT) technologies and arrangements with smart metering capabilities, eliminating the need for middlemen and allowing for more effective use of renewable energy sources. However, these direct energy exchanges between prosumers can be susceptible to cyber-threats, which hinder secure and effective energy transactions while protecting privacy. To enable safe and seamless energy transactions among prosumers and the grid, the cyber-security of IoT devices should be of paramount significance as a possible solution. Therefore, this paper focuses on securing the energy transactions among prosumers facilitated by smart meters. It aims to address potential threats against data integrity, confidentiality, and availability from the prosumers' point of view and develop a comprehensive framework for securing energy transactions based on artificial intelligence (AI). The proposed structured roadmap not only identifies compromised trading data but also prevents prosumers from reacting to it by replacing the contaminated as well as missing trading data. A comparative analysis on AI-based algorithms indicates that decision tree (DT) outperforms support vector machine (SVM) and multi-layer perceptron (MLP) for the proposed framework to profile the corrupted trading data identification and categorization in order to provide effective outcomes. Additionally, the proposed framework adopts a deep learning (DL)-based model for the replacement of compromised trading data. All the numerical analyses, along with extensive simulation results, justify, the efficacy of the proposed framework.

1. Introduction

Conventional consumers are proactively participating in the local electricity markets (LEMs) with the explosion of distributed energy resources (DERs), and the prosumer concept is being introduced [1]. Distributed generators have improved energy usage flexibility at the local distribution level [2]. Energy storage systems have also played important roles in this progression as they enable sustainable energy scheduling [3]. The integration of electric vehicles (EVs) and controllable loads further facilitates energy usage scheduling [4]. Hence, a smart energy management mechanism is essential to handle distributed generators, storage systems, EVs, and controllable loads in a coordinated way, as indicated in [5]. This smart energy management involves multiple operational stages to provide ultimate benefits to both prosumers and consumers. Bilateral energy trading can be conducted in one of the operational stages of smart energy management that enables prosumers and consumers to trade with each other and foster a more decentralized energy sharing [6,7].

With the widespread adoption of cutting-edge information and communication technologies (ICTs) and decentralized trading platforms, the bilateral energy trading-supported LEM paradigm has become a viable way to organize flexible energy trades among prosumers and consumers [6]. However, preserving the security of bilateral energy

* Corresponding author.
E-mail address: imran.azim@monash.edu (M.I. Azim).

https://doi.org/10.1016/j.segan.2024.101555
Received 4 March 2024; Received in revised form 10 September 2024; Accepted 20 October 2024
Available online 28 October 2024

Nomenclature	
$\mathscr{M}^{T}_{p,d,t}$	Trading data of a prosumer, $p \in \mathscr{P}$ on a particular day $d \in \mathscr{D}$ at any instance $t \in \mathscr{T}$
$\mathscr{M}^{D}_{p,d,t}$	Energy demand of a prosumer, $p \in \mathscr{P}$ on a particular day $d \in \mathscr{D}$ at any instance $t \in \mathscr{T}$
$\mathscr{M}^{G}_{p,d,t}$	Trading data of a prosumer, $p \in \mathscr{P}$ on a particular day $d \in \mathscr{D}$ at any instance $t \in \mathscr{T}$
$\widetilde{\mathscr{M}^{T}_{p,d,t}}$	Compromised trading data of a prosumer, $p \in \mathscr{P}$ on a particular day $d \in \mathscr{D}$ at any instance $t \in \mathscr{T}$
$\mathscr{N}(t)$	Outlier data
$\mathscr{B}_{FD}$	Bias factor for FD injection
$\mathscr{R}and(t)$	Random number at instance t
$\mathscr{T}_{afd}$	Attack period for FD injection
t_{st}	Starting time of the attack
t_{en}	Ending time of the attack
$\mathscr{B}_{Scan}$	Bias factor for scanning attack
$\mathscr{B}_{XSs}$	Bias factor for XSs attack
$\mathscr{B}_{BDoor}$	Bias factor for BDoor attack
$\mathscr{B}_{Ransom}$	Bias factor for Ransom attack
ξ	The instant at which the intruder records the trading data
$\mathscr{M}^{T}_{i}$	Dataset of ith prosumer
$\mathscr{M}^{T}_{i\_Comp}$	Compromised part of dataset $\mathscr{M}^{T}_{i}$
$\mathscr{M}^{T}_{i\_ben}$	Benign part of dataset $\mathscr{M}^{T}_{i}$
$\mathscr{M}^{T}_{i\_FD}$	Corrupted data having FD injection attack
$\mathscr{M}^{T}_{i\_Scan}$	Corrupted data having scanning attack
$\mathscr{M}^{T}_{i\_XSs}$	Corrupted data having XSs attack
$\mathscr{M}^{T}_{i\_BDoor}$	Corrupted data having backdoor attack
$\mathscr{M}^{T}_{i\_DDoS}$	Corrupted data having DDoS attack
$\mathscr{M}^{T}_{i\_pass}$	Corrupted data having password attack
$\mathscr{M}^{T}_{i\_Ran}$	Corrupted data having ransomware attack
x	sample of the attribute
y	True label of the attribute, x
μ	mean
$\mathscr{K}$	Number of attributes' classes
σ_{SD}	standard deviation
$\mathscr{M}^{T}_{i\_Train}$	Training dataset of DT model for i_{th} prosumer
$\mathscr{M}^{T}_{i\_Test}$	Testing dataset of DT model for i_{th} prosumer
$\mathscr{G}$	Gini impurity of DT model
i_t	Input gate activation of LSTM at time t
σ_{LSTM}	Sigmoid activation at time t for LSTM
W_i	Input gate weight matrix of LSTM
h_{t-1}	Output for the prior time step
X_t	Input for time step t
B_i	Bias vector for input gate of LSTM
f_t	Forget gate activation of LSTM at time t
W_f	Forget gate weight matrix of LSTM
B_f	Bias vector for forget gate of LSTM

o_f	Output gate activation of LSTM at time t
W_o	Output gate weight matrix of LSTM
B_o	Bias vector for output gate of LSTM
$\widetilde{C}_t$	Candidate cell state
C_t	Cell state
h_t	Hidden state
$tanh$	Hyperbolic tangent activation function
$\odot$	Element-wise multiplication
P	Number of inputs of ANN
Q	Number of hidden nodes of ANN
f	Activation function of ANN
w_i	Weights from the input to the hidden nodes
w_j	Weights from the hidden layer to the output
a_j	The true value at the index point i
f_j	The predicted value for index point i
m	The total number of observations
x_{Prcn}	Precision of the testing data for a prosumer
x_{Rcl}	Recall of the testing data for a prosumer
$x_{F1\ Scr}$	F1 Score of the testing data for a prosumer
x_{Fpr}	False positive rate of the testing data for a prosumer
x_{Fnr}	False negative rate of the testing data for a prosumer

Abbreviations

IoT	Internet of Things
DERs	Distributed energy resources
SMs	Smart meters
XSs	Cross-site scripting
BDoor	Backdoor
FD	False data
ID	Intrusion detection
IPs	Internet Protocols
DDoS	Distributed Denial of Service
DT	Decision tree
ML	Machine learning
LSTM	Long short-term memory
ANN	Artificial neural network
DL	Deep learning
RNN	Recurrent neural network
ReLU	Rectified Linear Unit
MAE	Mean absolute error
RMSE	Root mean square error
SMAPE	Symmetric mean absolute percentage error
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
FPR	False Positive Rate
FNR	False Negative Rate

transactions is essential, especially when it comes to the scenario of data sharing using internet of things (IoT) devices. This is because the number of interconnected devices increases frequently on IoT platforms, leading to security risks and vulnerabilities that are associated with them [8]. Before using the bilateral energy trading concept in a practical decentralized database-based system, such as blockchain [9], various technical difficulties need to be addressed, including the problem of securing energy trade [10].

In order to collect data on appliance energy usage, broadcast information about energy shortages or surpluses, and provide clients with pricing information, advanced metering infrastructure is an indispensable part of real-time energy trading [11]. The home energy

management system (HEM) uses smart meters (SMs) to regularly aggregate the energy data from all electrical appliances (EAs) in order to achieve the goal of successful trading. SMs enable prosumers to make wise decisions about energy transactions by giving them access to real-time knowledge about energy supply, pricing, and usage patterns [12].

Additionally, IoT technology makes home appliances smarter, enhancing the functionality of HEMs. By enabling real-time monitoring and two-way communication between energy providers and users, SMs have revolutionized the energy sector. Without SMs, it would have been difficult or impossible for the distribution system operator to deploy a number of services [13]. However, the enhanced connectivity and wireless communication capabilities of SMs also bring forth certain security flaws. This wireless communication route may be open to unauthorized intrusion, allowing malevolent opponents the opportunity to intercept or alter data and obtain access to confidential data [14]. Data integrity, confidentiality, availability, and accountability are the four categories into which the cyber security problems originating from the flaws in SMs' communication architecture can be divided [15].

In SM, an integrity attack is a hostile act intended to jeopardize the integrity of trading data or critical system information. Lack of integrity may lead to shared data being modified or altered to undermine its accuracy, dependability, or trustworthiness [16]. Attacks against secrecy focus on unauthorized parties' access to or disclosure of private and proprietary information. The most significant sources of the secret breaches in bidirectional energy transactions are the SMs [17]. Through root password recovery or security imperfections, the adversary can access SMs [18] and gather data on consumers' electricity usage in order to violate their privacy.

For the effective and steady operation of the smart grid, data availability is also crucial. This enables prompt and dependable access to information. Cyberattacks on availability disrupting data transfers can delay, block, or even distort the control signal, which has a significant negative influence on the stability, effectiveness, and security of the smart grid operation [19]. Because of this broad spectrum of threats, it is crucial to safeguard the prosumer energy trading process, which utilizes the IoT platform [20]. To this end, a defense plan against the false data (FD) attack on the demand-response program and peer-to-peer energy trading is discussed by the authors in [21,22]. A distributed denial of service (DDoS) attack prevention method for advanced metering equipment is proposed in [23]. The authors in [24] propose a DDoS detection scheme based on DL. However, there is no DDoS mitigation strategy for either of the algorithms. Without any attack mitigation strategies, an algorithm for detecting and preventing ransomware is presented in [25]. The authors in [26] provide a password attack detection scheme to ensure the confidentiality of the data. Strategies to recognize and prevent scanning (Scan), cross-site scripting (XSs), and DDoS attacks are provided in [27] and [28], with an emphasis on the significance of preserving data availability and integrity. However, the mitigating strategy is not available. A backdoor (BDoor) attack defense method is reported in [29], which merely provides the attack's detection scheme. In many spheres of society, including banking [30], education [31], medical [32], the manufacturing sector [33], and particularly in the area of cyber security [34], the use of artificial intelligence (AI) approaches is expanding dramatically. They are now being used by the modern power industry for a variety of purposes, including load forecasting, economical load dispatching, and energy management [35].

Motivated by the effectiveness and low time-consuming characteristics of AI-based schemes, this paper aims to open the door for the widespread adoption of secure two-way energy trading, enabling prosumers to actively participate in the energy market while maintaining the confidentiality of their data and the integrity of energy transactions. Moreover, the majority of earlier literature focuses on data integrity, availability, and confidentiality issues independently, and most algorithms are only used for attack detection. Some of them have some degree of ability to defend against cyberattacks like FD injection. However, in addition to the detection and prevention of cyberattacks, there should be a strategy that can mitigate the effects of the attack in order to maintain a stable energy market. To the best of the authors' knowledge, there is currently no cyber-secured solution for SMs in bidirectional energy transactions that can guarantee data integrity, confidentiality, and unavailability by combining attack detection, prevention, and impact mitigation. Although some previous literature is confined to attack detection and prevention to some extent, no work is proposed focusing on attack impact mitigation through replicating the original data. Here lies the unique contribution of this work whose emphasis is placed on filling this research gap that can identify, prevent, and mitigate the effects of attacks while maintaining the integrity, unavailability, and confidentiality of SMs data with the use of an AI-based protection scheme. This approach uses synthetic data that directly reflects exchanged information amongst prosumers. Prosumer data will have a significant effect on the choices made during energy trading and support the analysis of the energy market. An overview of different security approaches adopted by existing studies, along with the proposed AI-based scheme, is summarized in Table 1. This research contributes to the creation of a strong and reliable energy ecosystem by tackling the security issues related to smart meters by detecting, preventing, and mitigating the impact of attacks with the help of an AI-based scheme. Here, decision tree (DT), a machine learning (ML)-based algorithm, is used to determine whether shared data is secure or compromised before classifying it into various attack types, including FD injection, Scan, XSs, BDoor, DDoS, password, and ransomware attacks.

When compromised trading data is discovered, the model alerts the controlling entity and diverts it to a separate folder called the "spam folder" to prevent the prosumer's response to it. The original trading data is then retrieved using a deep learning (DL)-based method in order to diminish the impact of the attack. Here, the distorted data is compensated using a long short-term memory-based technique combined with an artificial neural network (LSTM-ANN). The main contributions of this paper, focusing on the development of an extensive framework for the security of prosumer energy transactions and the protection of smart meter data, are as follows:

- A cyber-secured framework is proposed to detect and mitigate cyber threats by utilizing prosumers' energy transaction data to facilitate the decisions taken during energy trading.
- The dataset is meticulously assembled with diverse attack variants, namely FD injection, Scan, XSs, BDoor, DDoS, Password, and Ransomware, in the proposed model to target vulnerable prosumers while taking into account outlier data from SMs to train the AI-based model.
- The proposed model takes action to prevent prosumers from reacting to compromised transaction data, and the compromised data is replaced by forecasted data.
- An extensive performance evaluation is performed for both ML-based attack identification and DL-based data forecasting schemes, along with a data security, perspective to verify the efficacy of the proposed model.
- The proposed model's effectiveness in improving the security and integrity of energy transactions is further demonstrated by the numerical analyses of energy transaction data.

The arrangements for the rest of this paper are as follows. Section 2 presents the energy transaction framework in a summarized manner, along with its vulnerability. The details of potential attack models and the proposed framework to handle those models are described in Section 3 and Section 4, respectively. The performance evaluation through comparative analysis is given in Section 5. Lastly, conclusions are drawn in Section 6.

Table 1
Overview of security approaches for different cyber assaults.

Literature	Data integrity	Data unavailability	Data confidentiality	Approaches		
				Detection	Prevention	Impact Mitigation
[21]	FD injection	×	×	✓	×	×
[22]	FD injection	×	×	✓	×	×
[23]	×	DDoS	×	×	✓	×
[24]	×	DDoS	×	✓	×	×
[25]	×	×	Ransom	✓	✓	×
[26]	×	×	Password	✓	×	×
[27]	Scan	DDoS	×	✓	✓	×
[28]	XSs	DDoS	Password	✓	✓	×
[29]	BDoor	×	×	✓	×	×
Proposed AI-based scheme	FD injection Scan XSs BDoor	DDoS	Password Ransom	✓	✓	✓

[here, ✓and ×indicate the presence and absence in literature respectively]

2. Overview of energy transaction framework for attack implementation

Secured energy trading amongst prosumers on a completely decentralized and reliable platform, for example, a blockchain database [36]. Undoubtedly, a decentralized system can be the victim of outside attacks. The situation could be intensified if prosumers start sharing their data for the purpose of bilateral energy transactions. In this work, the trade-worthy net metering data of the prosumers is considered as the target of the adversaries. Based on the manipulated metering data, all the trading parameters can be corrupted during cyberattacks. In a general trading system, the trade-worthy data $\mathscr{M}^{T}_{p,d,t}$ shared by a prosumer $p \in \mathscr{P}$ on a particular day $d \in \mathscr{D}$ at any time instance $t \in \mathscr{T}$ can be represented as:

$$\mathscr{M}^{T}_{p,d,t} = \mathscr{M}^{D}_{p,d,t} - \mathscr{M}^{G}_{p,d,t}, \quad \forall p \in \mathscr{P}, \forall d \in \mathscr{D}, \forall t \in \mathscr{T} \tag{1}$$

where $\mathscr{M}^{G}_{p,d,t}$ represents the energy generation of a prosumer $p \in \mathscr{P}$ using its available DERs and $\mathscr{M}^{D}_{p,d,t}$ is the energy demand of that prosumer at a particular instant $t \in \mathscr{T}$. Based on the energy production and consumption of prosumer p, the following criteria for $\mathscr{M}^{T}_{p,d,t}$ can be obtained:

$$\mathscr{M}^{T}_{p,d,t} < 0, \text{ if } \mathscr{M}^{D}_{p,d,t} < \mathscr{M}^{G}_{p,d,t}, \quad \forall p \in \mathscr{P}, \forall d \in \mathscr{D}, \forall t \in \mathscr{T} \tag{2}$$

$$\mathscr{M}^{T}_{p,d,t} > 0, \text{ if } \mathscr{M}^{D}_{p,d,t} > \mathscr{M}^{G}_{p,d,t}, \quad \forall p \in \mathscr{P}, \forall d \in \mathscr{D}, \forall t \in \mathscr{T} \tag{3}$$

$$\mathscr{M}^{T}_{p,d,t} = 0, \text{ if } \mathscr{M}^{D}_{p,d,t} = \mathscr{M}^{G}_{p,d,t}, \quad \forall p \in \mathscr{P}, \forall d \in \mathscr{D}, \forall t \in \mathscr{T} \tag{4}$$

where (2) indicates the phenomenon when a prosumer $p \in \mathscr{P}$ has spare energy to sell to the grid or other prosumers and the meter shows negative trading data. Under this condition, the prosumer acts as a seller. A positive value of metering data, $\mathscr{M}^{T}_{p,d,t}$, in contrast, indicates a shortage in energy generation, and the prosumer, p, has to purchase energy from the grid or other prosumers, as shown in (3). On the other hand, when the generated energy at any time $t \in \mathscr{T}$ is sufficient to satisfy the demand at that time, the trading condition depicted in (4) arises, i.e., $\mathscr{M}^{T}_{p,d,t} = 0$ and the prosumer, $p \in \mathscr{P}$, is neither a seller nor a buyer. During cyberattacks, attackers can manipulate $\mathscr{M}^{T}_{p,d,t}$ to get some financial gains deceptively as well as cause severe damage to the physical infrastructure.

3. Potential attack models

While data sharing paves the way to execute bilateral transactions among prosumers, several potential cyberattacks can be associated with it. This section outlines different types of threats that can be barriers to precise energy transactions. As an illustration, how these attacks affect a bidirectional energy transaction among prosumers is shown in Fig. 1. From each threat category, some are summarized here, which are used later to investigate the effectiveness of the proposed framework.

3.1. Data integrity attack

The unlawful and covert modification, alteration, or destruction of shared data results in the integrity loss of an energy market, and SMs should be able to fend off these kinds of attacks. When it comes to integrity, the majority of the literature is directed toward the mathematical modeling of fraudulent data injection. However, attacks like Scan [27], XSs [37], and BDoor [38] are increasingly prevalent in IoT-enabled devices and can be used to weaponize all modern IoT devices with botnets defined as networks of compromised equipment controlled remotely by attackers that allow them to carry out different malicious activities. This work models the Scan, XSs, and BDoor attacks in addition to the integrity attack by FD.

3.1.1. False data (FD) injection attack

This form of assault is capable of jeopardizing the most important concern of data integrity by infecting devices and bypassing firewalls. The information that is transmitted and the decisions that are necessary to sustain a healthy energy trading system can be manipulated through FD injection [22]. Pulse, scaling, ramp, random, smooth curve, and blind threats are frequently used for simulating the integrity attack by the FD injection [39]. Under the pattern of random attack template as given in [39], the compromised trade-worthy data $\widetilde{\mathscr{M}^{T}_{p,d,t}}$ can be represented as:

$$\widetilde{\mathscr{M}^{T}_{p,d,t}} = \begin{cases} \mathscr{M}^{T}_{p,d,t} + \mathscr{N}(t), & \forall t \notin \mathscr{T}_{afd} \subset \mathscr{T} \\ \mathscr{M}^{T}_{p,d,t} + \{\mathscr{B}_{FD} \times \mathscr{R}and(t)\} + \mathscr{N}(t), & \forall t \in \mathscr{T}_{afd} \subset \mathscr{T} \end{cases} \tag{5}$$

where $\mathscr{T}_{afd} \subset \mathscr{T}$ is the attack period under which the FD injection takes place. If $t_{st} \in \mathscr{T}_{afd}$ is the starting time and $t_{en} \in \mathscr{T}_{afd}$ is the end time of the attack then $\mathscr{T}_{afd} = [t_{st}, \dots, t_{en}]$. $\mathscr{R}and(t), t_{st} < t < t_{en}, \forall t \in \mathscr{T}$, generates random number and can be represented as:

$$\mathscr{R}and(t) = \begin{bmatrix} r_1 \\ r_2 \\ . \\ . \\ . \\ r_n \end{bmatrix} \tag{6}$$

where $r_1, r_2, \dots, r_n$ are the elements of $\mathscr{R}and(t)$ at tth, $(t+1)$th, $\dots$, $(t+n-1)$th instants, respectively, and $t, (t+1), \dots, (t+n-1) \in \mathscr{T}_{afd}$. $\mathscr{B}_{FD}$ represents the bias factor used by the intruder for FD injection. $\mathscr{N}(t), \forall t \in \mathscr{T}$, in (5) represents the outlier data that inherently existed along with the metering data. The details about this bad data are provided in Section 4.

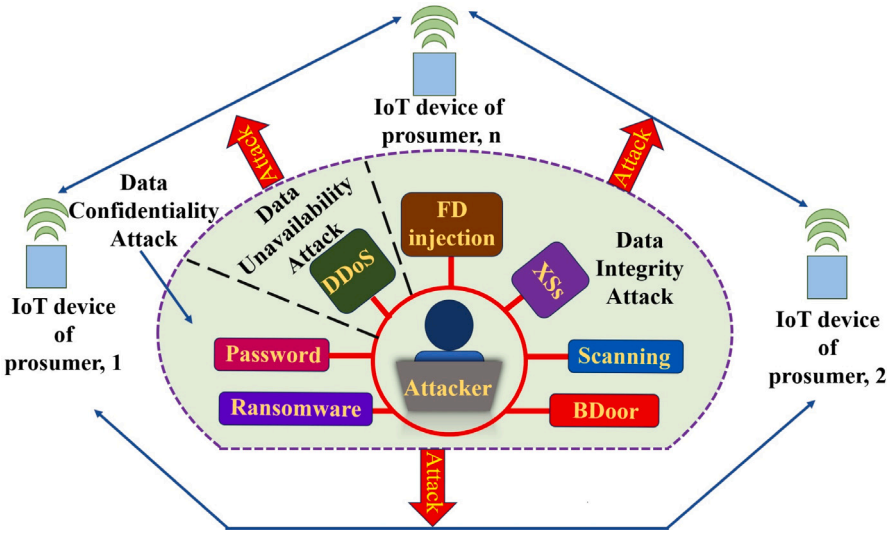


Fig. 1. An illustration of cyber attacks on energy transaction.

3.1.2. Scanning (scan) attack

Attackers strategically employ clever tactics to keep up with rapidly evolving technologies. They switch to multi-step attacks rather than single-stage attacks or low-level attacks [40]. The initial step in these kinds of multi-step attacks is to gather essential data about their intended victims. The initial access "scanning" phase, during which an attacker begins testing open ports on hacked IoT devices with scanner internet protocols (IPs), is one of the most crucial attack stages in an IoT attack life cycle. The attack can be started remotely by taking advantage of a remote code execution vulnerability. The scan attack, which is regarded as the first step in several multi-step attacks [41], yields important details on the targeted victims within the connected network. While doing so, it captures a lot of data that are required for the attack's later stages.

The Scan attack can result in network congestion, mostly harming network-dependent services. Customers' logins to the management webpage may also experience poor response times and occasionally browser timeouts. Energy transactions may suffer if the network delay goes above a particular level because services may offer erroneous data. Considering the Scan attack strategy as in [27], it is anticipated that the attackers have selectively altered trade-worthy metering data by reducing their values by a particular percentage in order to mimic the Scan attack. Under this consideration, the compromised trade-worthy data $\widetilde{\mathscr{M}^{T}_{p,d,t}}$ of a particular prosumer, $p \in \mathscr{P}$ on a particular day $d \in \mathscr{D}$ at any time instant $t \in \mathscr{T}$, can be represented as:

$$\widetilde{\mathscr{M}^{T}_{p,d,t}} = \begin{cases} \mathscr{M}^{T}_{p,d,t} + \mathscr{N}(t), & \forall t \notin \mathscr{T}_{afd} \\ \mathscr{M}^{T}_{p,d,t} - \{\mathscr{B}_{Scan} \times \mathscr{M}^{T}_{p,d,t}\} + \mathscr{N}(t), & \forall t \in \mathscr{T}_{afd} \end{cases} \tag{7}$$

where $\mathscr{B}_{Scan}$ denotes the bias constant used by the intruder for corrupting the SM data and $\mathscr{N}(t), \forall t \in \mathscr{T}$, is the outlier data generated while accumulating the trading information.

3.1.3. Cross-site scripting (XSs) attack

XSs attacks involve the injection of malicious code into a web page, which the victim's browser subsequently executes when they access the website. Attacks that exist in the applications used by the IoT device to communicate with the user are primarily caused by vulnerabilities in software [42]. In an IoT scenario, the problem worsens since an attacker can utilize the device credentials they have obtained to acquire root access to the device following a successful execution. This allows the attacker the ability to distribute malicious software updates that alter functionalities and use the infected device to carry out other kinds of large-scale attacks.

In the case of SMs, an attacker may insert malicious code into the online interface that enables prosumers to examine their energy usage data. The malicious code could intercept a user's login information or send information about their energy use to a remote server under the attacker's control when they visit the page. It is also possible for the attacker to inject malware that crashes the SM, resulting in inaccurate readings or possibly actual harm to the device. If the meter is a component of a wider energy system, there may be possible safety risks as well as financial losses for the end-user. Following the attack strategy as in [43] for generating the compromised data having XSs vulnerability, the corrupted net metering data of a specific prosumer $p \in \mathscr{P}$ is as follows:

$$\widetilde{\mathscr{M}^{T}_{p,d,t}} = \begin{cases} \mathscr{M}^{T}_{p,d,t} + \mathscr{N}(t), & \forall t \notin \mathscr{T}_{afd} \\ \{\mathscr{M}^{T}_{p,d,t} \times \mathscr{B}_{XSs}\} + \mathscr{N}(t), & \forall t \in \mathscr{T}_{afd} \end{cases} \tag{8}$$

where $\mathscr{B}_{XSs}$ refers to the bias factor used by the intruder for corrupting the SM data and $\mathscr{N}(t), \forall t \in \mathscr{T}$ is the bad data that exists inherently within the trading information.

3.1.4. Backdoor (BDoor) attack

Although there are some security standards for IoT devices such as ETSI Standards: EN 303 645, NIST cybersecurity framework, ISO/IEC 27001 [44,45], IoT devices can be vulnerable to attacks. Insufficient supplier application of security standards, default or weak settings, a lack of upgrades for legacy systems, unsafe remote access features, dependence on third-party components, vulnerability to supply chain assaults, and the complexity of networked IoT ecosystems are all potential factors of cyber-attacks. Backdoor attacks, a sort of poisoning attack, can target vulnerable IoT devices [46]. An SM BDoor attack entails an attacker getting unauthorized access to the meter's software and installing a BDoor that gives them control over the meter from afar. As a result, the attacker may be able to influence the data the meter is gathering or even take over its operation. The attacker could, for instance, alter the meter readings to make it seem like less energy is being used than it actually is. This could lead to the client paying less than they should for their energy use, which would cause the energy provider to lose money.

A BDoor can enable an attacker to manipulate the smart meter in addition to altering the data being collected. This can entail cutting off the power to the client's residence or place of business or even physically harming the meter. It is assumed that the poisoning-based BDoor attack scheme as in [47] is chosen by the attacker, which has poisoned the original data with the predefined triggered value, i.e., the

bias constant set by the attacker. Under this pattern, the compromised trade-worthy data $\widetilde{\mathscr{M}^T_{p,d,t}}$ of a particular prosumer, $p \in \mathscr{P}$ on a particular day $d \in \mathscr{D}$ at any instance $t \in \mathscr{T}$, can be represented as:

$$\widetilde{\mathscr{M}^T_{p,d,t}} = \begin{cases} \mathscr{M}^T_{p,d,t} + \mathscr{N}(t), & \forall t \notin \mathscr{T}_{afd} \\ \{\mathscr{M}^T_{p,d,t} \times \mathscr{B}_{BDoor}\} + \mathscr{N}(t), & \forall t \in \mathscr{T}_{afd} \end{cases} \tag{9}$$

where $\mathscr{B}_{BDoor}$ and $\mathscr{N}(t), \forall t \in \mathscr{T}$, represent the bias factor used by the adversarial to poison the original data and the outlier data presented in the system, respectively.

3.2. Data unavailability attack

Data availability ensures prompt and trustworthy access to information, which is crucial for the effective and steady operation of the energy market. Cyberattacks on availability that disrupt data flows can postpone, hinder, or even distort information and have a significant negative influence on the stability, effectiveness, and security of the operation of the energy market. Attacks on data availability aim to render the victim unusable for legitimate services by depleting the victim's essential resources, including CPU, memory, bandwidth, and server buffer capacity. Because of its detrimental impact on the IoT framework and energy trade, DDoS, one of the Botnet attacks that target IoT devices, has recently gained a lot of attention [48,49]. Under this attack, it is assumed that the attacker makes the victim prosumer $p \in \mathscr{P}$ on a particular day $d \in \mathscr{D}$ at any time instance $t \in \mathscr{T}$ unavailable and then the manipulated trade-worthy data of the victim prosumer can be represented as:

$$\widetilde{\mathscr{M}^T_{p,d,t}} = \begin{cases} \mathscr{M}^T_{p,d,t} + \mathscr{N}(t), & \forall t \notin \mathscr{T}_{afd} \\ \mathscr{N}(t), & \forall t \in \mathscr{T}_{afd} \end{cases} \tag{10}$$

3.3. Data confidentiality attack

One of the main sources of the confidential data leak in the smart grid is the SM [50]. By allowing unauthorized access to SM data, a confidentiality assault results in the disclosure of consumers' private and personal information. Aside from password attacks, IoT devices can become an appealing target for ransomware, which can have a significant impact on energy transactions [51].

3.3.1. Password attack

All IoT devices are operated by individuals who are prone to mistakes. Keeping default passwords or selecting weak passwords is one of the most common blunders. Using tools such as dictionaries or probabilistic models, hackers can get access to SMs via root password recovery. These tools can significantly lower the number of tries required to guess a password and collect consumer electricity usage data to violate customers' privacy. For preparing the corrupted net metering data, it is assumed that the attacker records an arbitrary number of trading data for a prosumer $p \in \mathscr{P}$, and echoes the recorded data instead of reporting the new trade-worthy data. Under this consideration, following the attack pattern as in [52], the manipulated data can be represented as:

$$\widetilde{\mathscr{M}^T_{p,d,t}} = \begin{cases} \mathscr{M}^T_{p,d,t} + \mathscr{N}(t), & \forall t \notin \mathscr{T}_{afd} \\ \{\mathscr{M}^T_{p,d,(t-\xi)} + \mathscr{N}(t-\xi) + \mathscr{N}(t), & \forall t \in \mathscr{T}_{afd} \end{cases} \tag{11}$$

where $\xi, \forall \xi \in \mathscr{T}$ and $0 < \xi < t$ indicate the instant at which the intruder records the trading data and exchanges it with the data at tth instant.

3.3.2. Ransomware attack

Until a ransom is paid to the attacker, ransomware can block a user from reaching a device and its files. The main goal of ransomware is to encrypt private files and demand significant sums of money to unlock them. In certain cases, the attacker will threaten to release the private files to the public if the ransom is not paid. The inclusion of vulnerable IoT devices in the energy market could have a disastrous financial impact on energy trade [53]. Energy trading platforms are vulnerable to attacks of this nature, which might have major repercussions for both the platform and the larger energy infrastructure. This assault may prevent the platform from operating normally and may cause the loss of sensitive data, including customer and transaction information. Additionally, this kind of attack may cause clients to have doubts about the dependability and security of energy trading systems. In addition to these issues, a ransomware assault could disrupt the grid's operation, leading to power outages and other issues that could also impair the platform's functionality. For designing the corrupted trade-worthy data under a ransomware attack, this work considers the situation where the intruder wants to manipulate the data by adding a constant bias factor to each data point, so the manipulated trading data can be represented as:

$$\widetilde{\mathscr{M}^T_{p,d,t}} = \begin{cases} \mathscr{M}^T_{p,d,t} + \mathscr{N}(t), & \forall t \notin \mathscr{T}_{afd} \\ \{\mathscr{M}^T_{p,d,t} + \mathscr{B}_{Ransom}\} + \mathscr{N}(t), & \forall t \in \mathscr{T}_{afd} \end{cases} \tag{12}$$

where $\mathscr{B}_{Ransom}$ and $\mathscr{N}(t), \forall t \in \mathscr{T}$, indicate the bias factor used by the adversarial to poison the original data and the outlier data presented in the system, respectively.

4. Proposed framework

First, a synthetic dataset that takes into account all threat models has been created and preprocessed. To train the AI models; support vector machine (SVM), multi-layer perceptron (MLP), and DT; for identifying and categorizing the adversarial varieties, preprocessed data is utilized. The framework's first stage is to detect compromised data. So, if the model can detect corrupted trading data, then this data can be replaced with the predicted one. After selecting the most effective attack detection strategy, the DL-based model is trained using a dataset that only contains information about permissible trade in order to predict the actual data when any compromised data is received and replace fake data with the predicted one. The following subsections cover each phase in depth.

4.1. Artificial intelligence (AI)-based attack identification scheme:

4.1.1. Preparing dataset considering all attack patterns

To train the AI-based models for attack detection purposes, first of all, the dataset is generated considering all of the attack templates. Let $p = \{p_1, p_2, \ldots, p_n\}, \forall p \in \mathscr{P}$, be all prosumers' sets, including both the victims and safe prosumers. p_i is the prosumer index, where $i = 1, 2, \ldots, n$ and $\mathscr{M}^T_i$ indicates the dataset of ith prosumer. Each prosumer keeps the dataset $\mathscr{M}^T_i$ on the device locally. Dataset, $\mathscr{M}^T_i$, has two parts, namely: a compromised part, covering all the attack categories, $\mathscr{M}^T_{i\_Comp}$ and a benign part, $\mathscr{M}^T_{i\_Ben}$, such that:

$$\mathscr{M}^T_i = \mathscr{M}^T_{i\_Comp} \cup \mathscr{M}^T_{i\_Ben} \tag{13}$$

$\mathscr{M}^T_{i\_Comp}$ contains the trading data used for model training purpose considering all of the above-mentioned attack profiles, that is:

$$\mathscr{M}^T_{i\_Comp} = \mathscr{M}^T_{i\_FD} \cup \mathscr{M}^T_{i\_Scan} \cup \mathscr{M}^T_{i\_XSs} \cup \mathscr{M}^T_{i\_DDoS} \cup \mathscr{M}^T_{i\_BDoor} \cup \mathscr{M}^T_{i\_Pass} \cup \mathscr{M}^T_{i\_Ran} \tag{14}$$

here, $\mathscr{M}^T_{i\_FD}, \mathscr{M}^T_{i\_Scan}, \mathscr{M}^T_{i\_XSs}, \mathscr{M}^T_{i\_DDoS}, \mathscr{M}^T_{i\_BDoor}, \mathscr{M}^T_{i\_Pass}, \mathscr{M}^T_{i\_Ran}$ indicates the corrupted data having FD injection, Scan, XSSs, BDoor, DDoS,

Table 2
Attack strategies.

Security breach	Attack types	Targeted days by the intruders for each prosumer						Corrupted data of individual attack
		5th	9th	11th	13th	22th	26th	
Integrity	FD Injection	21st	21st	21st	21st	21st	21st	3% of total data
	Scan	25th	23rd	22nd	16th	18th	26th	
	XSs	26th	26rd	23nd	29th	23th	30th	
	BDoor	12th	13th	6th	10th	5th	15th	
Unavailability	DDoS	16th	17th	9th	11th	8th	17th	
Confidentiality	Password	2nd	3rd	2nd	6th	2nd	6th	
	Ransomware	24th	20th	20th	13th	17th	24th	

[The total corrupted data for each prosumer considering all of the attacks is 21% of total data]

password, and ransomware attack templates, respectively. The trade-worthy metering dataset, $\mathscr{M}_i^T, \forall \mathscr{M}_i^T \in \mathscr{M}^T$, having $\mathscr{K}$ classes, can be represented as:

$$\mathscr{M}_i^T = [(x, y)^{(j)}]_{j=1}^m \tag{15}$$

with $x \in \mathscr{X} \subset \mathrm{R}^m$ denoting a sample of the attribute and $y \in \mathscr{Y} = \{1, 2, \ldots, \mathscr{K}\}$ having its true labels. The classification model will learn a function $f(x, \phi)$ with parameters ϕ to map the input to the label, $f : \mathscr{X} \to \mathscr{Y}$. For a test sample, the model is expected to correctly predict the benign sample, i.e., $f(x_{Ben}, \phi) = y_{Ben}, x_{Ben} \in \mathscr{M}_{i\_Test}^T$. Moreover, it is also desirable to predict the adversarial class for any input that contains compromised data: $f(x_{Comp}, \phi) = y_{Comp}, x_{Comp} \in \mathscr{M}_{i\_Test}^T$. Here, y_{Comp} symbolizes all of the adversary class labels, and x_{Comp} indicates their respective data point.

The compromised data for FD injection, scan, XSs, BDoor, DDoS, password, and ransomware attacks follow the equations (5), (7), (8), (9), (10), (11) and (12) respectively. In this study, 30 residential prosumers' synthetic net metering data are used. The data set is spanned for one month with a time resolution of every 30 min. So, the 24-hour scheduling period is broken into 48 time slots since the time is scheduled from $12am$ to $11 : 30pm$. As the model is pre-trained solely on the prosumers' real trading data and implemented attack patterns, it is independent on the occurrence time of any kind of compromised net metering data. When attack dynamics appear at the prosumer end that were not considered in the training data sets, as per the designed framework's working steps, the model will first separate it from the trading environment and accumulate it in another folder called "spam folder" and at the same time the corrupted value would be replaced by the predicted value. After filtering the corrupted data, the model will classify it and send feedback to the control center. 7 days among 30 days were selected randomly for implementing the attack templates. 3% of data from a prosumer's data set is selected for the purpose of compromising by each attack category, and in total, 21% of the data of each prosumer is compromised when all the templates are implemented. The attack strategies for the prosumers are given in Table 2. The designed attack templates manipulate the shared net metering data of the 6 prosumers with different scaling factors.

4.1.2. Data preprocessing

The generated data is then preprocessed for classification problems. The presence of outlier data is inherent in smart-grid data, including data on electricity consumption, asset management, etc. [54,55]. The performance and accuracy of SMs, which are made up of sophisticated circuits and sensors used for data collection, vary according to the frequency of data acquisition. The interference prevention capability of the gadgets leads to a lot of measurement inaccuracies. However, outlier data, like noise, is a significant factor in lowering data quality and can negatively impact the performance of data-driven models [56]. Hence, it is regarded as bad data. In this approach, it is considered that the bad data appears in the form of white noise that obeys a Gaussian distribution pattern with a zero mean, i.e., $\mu = 0$, and a standard deviation, i.e., $\sigma_{SD} = 0.001$. $\mathscr{N}(t), \forall t \in \mathscr{T}$, in Eq. (5) indicates the noise data that exists with the SM data and can be represented as:

$$\mathscr{N}(t) = \begin{bmatrix} n_1 \\ n_2 \\ . \\ . \\ . \\ n_n \end{bmatrix} \tag{16}$$

where, $n_1, n_2, \ldots, n_n$ is the element of $\mathscr{N}(t)$ at tth, $(t+1)$th, $\ldots$, $(t+n-1)$th instant, respectively, where $t, (t+1), \ldots, (t+n-1) \in \mathscr{T}$ having power spectral density $\mathscr{S}_{\mathscr{N}}(f) = {\sigma_{SD}}^2$, for all the frequency and any two elements of $\mathscr{N}(t)$ are statistically independent, i.e., n_1 and n_2 are uncorrelated for any $t \neq (t+1)$. As different types of attacks are being implemented in this work, there is a possibility of $\mathscr{M}_{i\_Comp}^T \cap \mathscr{M}_{i\_Ben}^T \neq \emptyset$. To tackle this condition, for which the model can predict the wrong label, the feature normalization technique is adopted here. Depending on the nature of the data and the characteristics of the model, a variety of feature normalization methods can be utilized. At this step, only the benign trade-worthy data is normalized by scaling it up by a constant factor. The actual trade-worthy data can be obtained by simply dividing the anticipated output of the model by the same constant. Then, the outlier data is incorporated into each prosumer's data.

4.1.3. Attack detection and categorization

In this work, the detection model not only detects the compromised data but also prevents a prosumer from responding to it, and the corrupted data is being stored for future analysis. When the IoT device of one prosumer receives any data from another prosumer, the proposed model first checks whether the shared data is benign or not based on the dataset by which the model has been trained. If the data is safe for trading, then the particular prosumer's IoT device displays directly on the trading environment's display. But if the data is compromised, then this data is first filtered out and accumulated in another folder, namely the spam folder. From this spam folder, feedback about the compromised data is sent to the authorized entity, i.e., the control center. So, the detection model not only detects the compromised data but also prevents a prosumer from responding to it, and the corrupted data are also stored for analysis in the future. After comparing the performance evaluation metrics of SVM, MLP, and DT, the suitable AI-based model is selected for attack detection and categorization purposes for the framework.

Support Vector Machine (SVM):

Motivated by the effectiveness of support vector machine (SVM) classification algorithms in various fields such as pattern recognition, computer vision, image analysis, and business intelligence [57,58], it is chosen here for attack detection purposes. SVM is a supervised learning method whose working premise relies on finding the ideal hyperplane that optimally separates the various classes in the input data space. It operates by encoding input data as points in a high-dimensional space, with each feature denoting a dimension. SVM training includes solving an optimization problem to identify the ideal hyperplane. This

Table 3
Evaluation metrics of SVM for attack detection.

Prosumer's identity	Identification accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
5th	86.81	83.38	86.81	84.27
9th	89.93	88.74	89.93	88.25
11th	86.12	86.14	86.12	85.31
13th	89.91	89.72	89.91	89.27
22nd	90.85	90.01	90.85	89.71
26th	90.54	89.86	90.54	89.38

Table 4
Evaluation metrics of MLP for attack detection.

Prosumer's identity	Identification accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
5th	86.81	88.02	86.81	86.10
9th	88.89	87.69	88.89	86.81
11th	86.46	83.91	86.46	84.31
13th	87.85	88.50	87.85	87.77
22nd	86.46	85.64	86.46	85.42
26th	88.19	89.99	88.19	86.18

optimization challenge tries to decrease classification error while maximizing the margin. Once the ideal hyperplane has been found during training, SVM can classify fresh data points based on which side of the hyperplane they belong.

During the training process of our classification task, the radial basis function (RBF) serves as the kernel function, mapping the input data into a higher-dimensional space. Gamma, an RBF kernel parameter, is used to govern a single training example. Higher gamma values result in more complex decision boundaries, and their value is set to 0.2 throughout the training procedure. The regularization parameter, $\mathscr{C}$, governs the balance between maximizing margin and lowering classification error. For our categorization task, we kept the value at 30. Table 3 shows the evaluation metrics of SVM for attack detection purposes of the victim prosumers. The definitions of evaluation metrics are given in Section 5.2.3.

Multi-layer Perceptron (MLP):

Because the multi-layer perceptron (MLP) has shown superior performance in classification tasks [59,60], this simple feedforward neural network (NN) is used in this study to detect compromised data. In MLP design, the input layer receives raw input data, and each neuron represents a feature of the input data. The first hidden layer's neurons compute a weighted sum of the information received from the input layer. After computing the weighted sum, an activation function is used to add nonlinearity to the model. Rectified linear unit (ReLU) is a popular activation function because of its simplicity and efficacy in dealing with the vanishing gradient problem. The activation function's output is used as an input for the next layer. This procedure of calculating the weighted total, applying an activation function, and forwarding the result to the next layer is repeated for each consecutive hidden layer until the output layer is encountered. The output layer calculates the model's final output. Each neuron in the output layer represents a probability or score associated with a specific class. The softmax activation function is widely employed in the output layer of multi-class classification tasks because it normalizes the output scores into probabilities that add up to one.

In our classification problem, the input layer has one neuron, and the first hidden layer has 64 neurons with the ReLU activation function. The second hidden layer consists of 32 neurons that also use the ReLU activation function. There are 32 neurons in the output layer that apply the softmax activation function. After designing the model architecture, it is compiled with the Adam optimizer, an adaptive learning rate optimization technique that adjusts the model's parameters based on the gradient of the loss function. The sparse categorical cross-entropy loss function is utilized here, which is appropriate for multi-class classification issues. The model's performance for 55 epoch is evaluated using measures such as accuracy, precision, recall, and F1 score, the values of which are listed in Table 4.

Decision Tree (DT):

This attack detection is also performed by the DT algorithm. The popularity of DT stems from its simplicity and ease of use [61]. It is a widely used predictive model which is well-recognized for its robustness, ability to be understood, and undeniable usefulness in a variety of applications [62]. Since it can handle both numerical and categorical data well and is a member of the supervised learning class of algorithms, DT is used to effectively solve classification problems [63]. Research works as in [64] and [65] show that DT can perform exceptionally well in tasks involving data classification. Moreover, DT comes with some built-in tools for managing the dataset's outliers. Rather than relying on absolute distances, It uses ranking or relative ordering of feature values as the splitting criterion, such as Gini impurity. Hence, outliers therefore have no appreciable impact on the decision boundary. Motivated by these characteristics, DT is chosen here for attack identification purposes. DT, the supervised machine learning method, is based on a recursive tree structure. There are three parts to it: a root node, a path, and a leaf node. An item or attribute is represented by the root node of a tree. The various values of the parent node are represented by each divergence path in the tree. The predicted category of the attribute is represented by the leaf node. "If-then" rules are another way to represent the resulting tree.

The evaluation metrics related to attack identification scheme by DT are in Table 5. Table 6 represents the comparison of attack identification accuracy for the three above-mentioned AI-based algorithms: SVM, MLP, and DT. As, DT depicts more effective performance among the compared AI-based schemes, this work chooses DT for attack detection and categorization purposes. The proposed structure for securing energy transactions through attack detection, prevention, and mitigation techniques is shown in full detail in Fig. 2.

Following is a step-by-step explanation of how the decision tree resolved the attack detection and classification issue:

Step 1: The generated and preprocessed data of ith prosumer having all the attack templates for model training and testing purposes is split into training, $\mathscr{M}^T_{i\_Train}$, and testing sets, $\mathscr{M}^T_{i\_Test}$, where, $\mathscr{M}^T_i = \mathscr{M}^T_{i\_Train} \cup \mathscr{M}^T_{i\_Test}$, having 80% and 20% of the total data, respectively. To ensure reproducibility in the train-test split process, the random seed was given a specific value.

Step 2: At this step, the DT model is designed to perform the attack detection and classification functions. As attack detection and categorization model should be less computationally complex and the targets of the dataset are categorical, for the purpose of measuring the impurity, Gini impurity is used here, which indicates the probability of misclassifying an observation. The training data for ith prosumer can be represented as:

$$\mathscr{M}^T_{i\_Train} = [(x_{i1}, y_{i1}), \ldots, (x_{im}, y_{im})], \quad y_j \in [1, \ldots, \mathscr{K}] \tag{17}$$

Table 5
Evaluation metrics of DT for attack detection.

Prosumer's identity	Identification accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
5th	97.79	98.19	97.79	97.82
9th	98.74	98.88	98.74	98.74
11th	96.21	96.60	96.21	96.29
13th	98.11	98.27	98.11	98.12
22nd	97.48	97.56	97.48	97.33
26th	99.05	99.13	99.05	99.06

Table 6
Compared identification accuracy of SVM, MLP and DT.

Prosumer's identity	Identification accuracy of SVM (%)	Identification accuracy of MLP (%)	Identification accuracy of DT (%)
5th	86.81	86.81	97.79
9th	89.93	88.89	98.74
11th	86.12	86.46	96.21
13th	89.91	87.85	98.11
22nd	90.85	86.46	97.48
26th	90.54	88.19	99.05
Average accuracy (%)	89.02	87.44	97.90

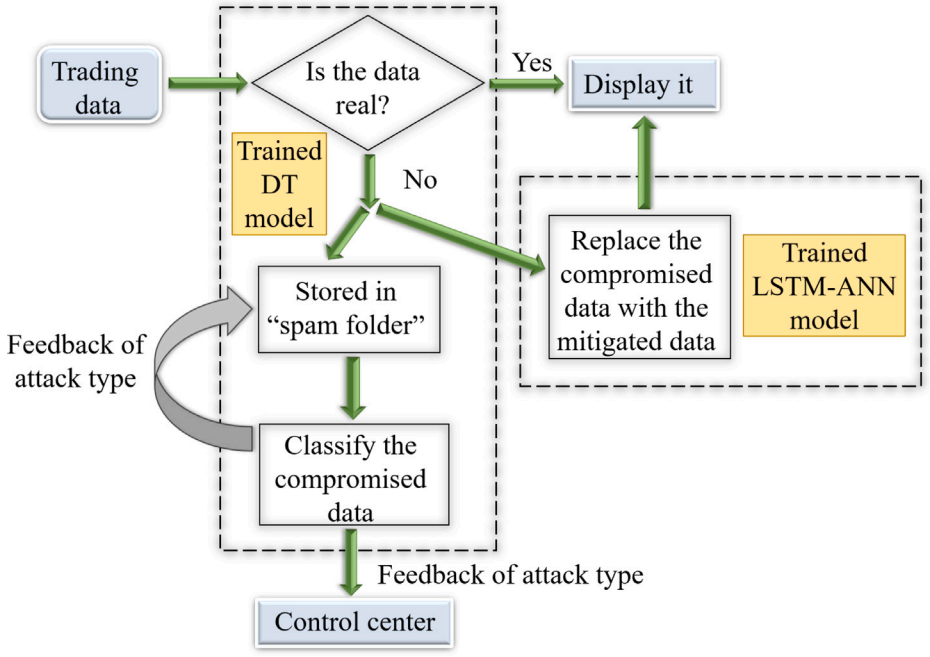


Fig. 2. A schematic illustration for securing trading data.

where $\mathscr{K}$ is the number of attributes' classes. If $(1-\mathscr{P}r_l)$ is the chance of not choosing a data point from class j and $\mathscr{P}r_l$ is the chance of choosing one, then the Gini impurity is:

$$\mathscr{G}(\mathscr{M}^T_{i\_Train}) = \sum_{j=1}^{\mathscr{K}}(\mathscr{P}r_l)\times(1-\mathscr{P}r_l) \tag{18}$$

Which can be simplified as:

$$\mathscr{G}(\mathscr{M}^T_{i\_Train}) = 1-\sum_{j=1}^{\mathscr{K}}(\mathscr{P}r_l)^2 \tag{19}$$

Except for the Gini criterion, the performance of DT for a classifying problem depends on the hyperparameters such as: $max\_depth, min\_samples\_leaf$ and $min\_samples\_split$.
How deep the tree can grow is determined by the max_depth parameter. By setting the max_depth to $\}\}none\epsilon$, the DT was created with the intention of ensuring that the tree would grow until all of its leaves were pure, or that all of the samples belonged to the same class. The minimal amount of samples that must exist in a leaf node is controlled by the $min\_samples\_leaf$ option. The leaf nodes contain a single sample when this parameter is set to $\}\}1\epsilon$ and for our problem this sample is the attack category. The split is skipped if there is a chance that a leaf node would have fewer samples than $min\_samples\_leaf$. To split an internal node, the minimum number of samples needed is determined by the $min\_samples\_split$ parameter. Having the configuration of $min\_samples\_split = 2$, the decision tree can split as soon as it notices even a slight difference between samples. It guarantees that all potential splits are explored by the model, which can be important for picking up on little patterns in the data.

Step 3: Then, the DT classifier is trained by feeding it the training data (x_{Train}) and the target values (y_{Train}) that correspond to them. The DT method discovers patterns and connections between the features and the target variable during training. Once trained, the model is then used to generate predictions based on fresh, unused data.

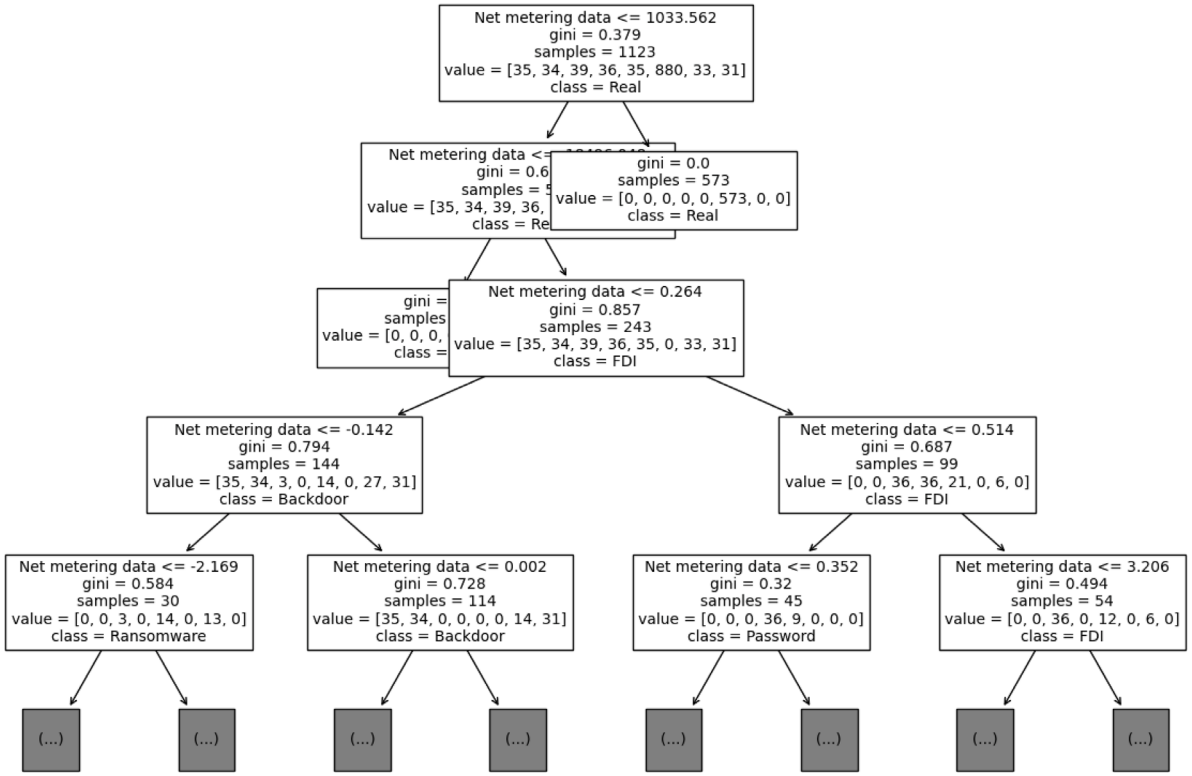


Fig. 3. Snippet model architecture of DT.

Table 7
Hyperparameters of the designed DT.

Hyperparameters	Value
Impurity	Gini
max_depth	*none*
min_samples_leaf	1
min_samples_split	2

Step 4: The effectiveness of the created DT model must be assessed following the completion of the training and testing phases. This may be accomplished for a classification issue by contrasting the expected labels with the actual labels. The next section has further information about the performance evaluation.

Table 7 summarizes the hyperparameters of the designed DT, and Fig. 3 shows the snippet of the DT model architecture starting from root node for attack detection purposes.

4.2. Deep learning (DL)-based attack effect mitigation

Upon detecting any compromised trading data, the framework's next step is to mitigate the attack impact by removing the corrupted data from the trading environment and replacing it with the predicted value. After comparing the performance evaluation metrics of long short-term memory (LSTM), and long short-term memory with artificial neural network (LSTM-ANN), the suitable AI-based model is selected for attack impact mitigation purposes for the framework.

4.2.1. Long short-term memory (LSTM)

Long Short-Term Memory (LSTM), defined as a variety of recurrent neural network (RNN) architectures, is frequently employed for sequential data processing [66]. To more effectively capture long-range relationships and address the vanishing gradient issue, LSTM adds memory cells and gating mechanisms. This characteristic makes them suitable for situations where historical context is required since it allows them to capture important data from far-past inputs. The key component of the LSTM network is the memory cell, which replaces the concealed layers of conventional neurons. The input, output, and forget gates of the LSTM network allow it to add and delete data from the cell state. As discussed in [67], the output of the LSTM can be computed as follows after updating the cell's state:

$$i_t = \sigma_{LSTM}(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{20}$$

$$f_t = \sigma_{LSTM}(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{21}$$

$$o_t = \sigma_{LSTM}(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{22}$$

$$\widetilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{23}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{24}$$

$$h_t = o_t \odot tanh(C_t) \tag{25}$$

The amount of fresh data that is added to the cell state at time step t is determined by the input gate activation, i_t. How much of the prior cell state is kept is determined by the forget gate activation f_t at time step t. The amount of the cell state that is exposed to the output is determined by o_t, which is the output gate activation at time step t. The input for time step t is x_t and the output (hidden state) of the prior time step is represented by the symbol h_{t-1}. Bias vectors are b_i, b_f, b_o, and b_C while weight matrices are W_i, W_f, W_o, and W_C. $\widetilde{C}_t, C_t$, and h_t represent the candidate cell state, cell state, and hidden state, respectively. The hyperbolic tangent activation function is known as $tanh$ while sigma (σ_{LSTM}) stands for sigmoid activation. $\odot$ represents element-wise multiplication and concatenation of h_{t-1} and x_t is denoted by the symbol $[h_{t-1}, x_t]$. These equations show the information flow through an LSTM cell, where the candidate cell state and hidden state both capture and transmit pertinent information, while the input gate, forget gate, and output gate control the flow of information into and

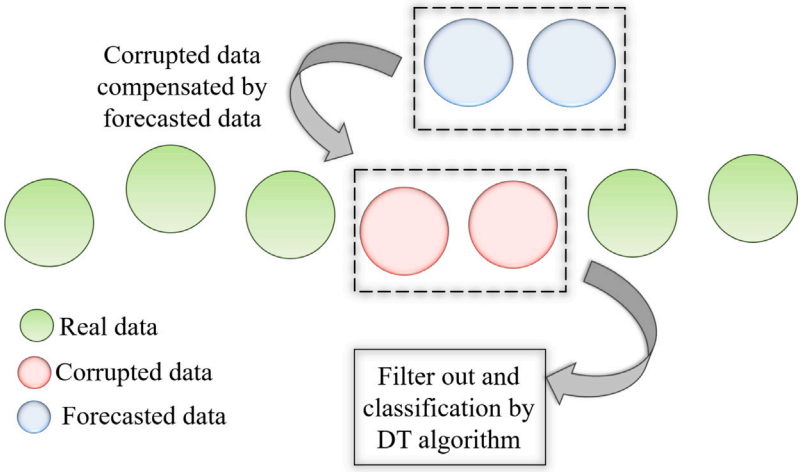


Fig. 4. A schematic illustration for compensating corrupted data.

Table 8
Predicting accuracy of the LSTM model.

Prosumer's identity	MAE	RMSE
5th	0.375	0.664
9th	0.066	0.049
11th	0.095	0.165
13th	0.035	0.043
22nd	0.107	0.198
26th	0.052	0.074

out of the cell state. Table 8 shows the evaluation metrics of LSTM for attack impact mitigation purpose of the victim prosumers. The definitions of evaluation metrics are given in Section 5.2.2.

4.2.2. Long short-term memory with artificial neural network (LSTM-ANN)

To compensate for the compromised trading data, the performance of a long short-term memory-based algorithm incorporated with an artificial neural network (LSTM-ANN) is also analyzed here. The researchers of [68] demonstrate that although LSTM is a commonly used algorithm for time series data prediction, there may still be opportunity for accuracy enhancement through the hybridization process. It was shown in [69,70] how an ANN architecture integrated with an LSTM performs better than a standalone LSTM model.

Furthermore, as demonstrated in [71], a robust architecture can be produced by combining a DL model based on LSTM-ANN. Motivated by the performances of LSTM-ANN, the combined architecture is used in this work for the purpose of mitigating the effect of corrupted data. The dataset, which has only the information of permissible trade data along with outlier data, is used to train this DL-based LSTM-ANN model. The process of mitigating the effect of corrupted data is shown schematically in Fig. 4, whereas Fig. 5 represents the data prediction process by the proposed LSTM-ANN.

ANN refers to the well-known artificial intelligence method that has the capacity to map data within the network's neurons. The brain's organic neurons serve as inspiration for the mechanism, which replicates their actions when processing input to generate predictions. ANNs are made up of interconnected artificial neurons, or nodes, arranged in three layers: input, hidden, and output. The input layer receives the features of the input data. One or more layers between the input and output layers are considered hidden layers, while numerous neurons make up each hidden layer. The final output predictions are produced by the output layer. As mentioned in [71], the output for the next node is created by preprocessing the input signals through these layers in accordance with the following equation and activation function:

$$y_t = w_0 + \sum_{j=1}^{Q} w_j \cdot f\left(w_{0j} + \sum_{i=1}^{P} w_{i,j} \cdot y_{t-1}\right) \tag{26}$$

where P and Q, respectively, are the number of inputs and hidden nodes, and f is the activation function. Weights from the input to the hidden nodes and from the hidden layer to the output are respectively, $w_{i,j}$ where $i = 1, 2, \ldots, P$ and $j = 1, 2, \ldots, Q$ and w_j, where $j = 0, 1, \ldots, Q$. w_{0j} stands for the weight for each output between the input and the hidden layer. The main factor in the widespread use of ANNs as a black-box model is their effectiveness in representing complex nonlinear relationships between target and input. However, the two-stake LSTM model is integrated here with it in order to improve predicted accuracy by utilizing the benefits of ANN (dense layers).

This work combines two LSTM layers with 256 and 128 neurons, followed by two dense layers with 100 and 50 units, to forecast energy transaction data. The *tanh* activation function is used by both the LSTM layer and can be computed, as shown in (27). In contrast, the dense layer employs a rectified linear unit (ReLU) that works, as demonstrated in (28). A final output layer is then created with the explicit goal of forecasting prosumer energy trade data in the future.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{27}$$

$$R(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

The temporal interdependence of the data is captured using a sliding window approach. It entails generating continuous input–output pairs by sliding a fixed-size window across the dataset. In this work, the window size is selected through trial and error as inputs to estimate the future value, as the security issue and the forecasting model's accuracy can be influenced by the window size [72]. For example, a window size needs to be large enough to capture the temporal characteristics of data, which assists in improving the accuracy of a forecasting model. On the other hand, using a large number of samples as the window size increases the possibility of more corrupted data within a window, which can affect the forecasting accuracy, even after being replaced by the forecasting values. Through trial and error, a window size of 30 samples is found to be the optimum for the selected model in this work. The sliding window approach creates a matrix relationship between input and output where the inputs serve as features of the model and the output is a function of input-lagged variables. The following equations describe how the entire process works [67].

$$IO = [XY] \tag{29}$$

$$X = \begin{bmatrix} x(1) & x(2) & \cdots & x(\tau) \\ x(2) & x(3) & \cdots & x(\tau+1) \\ \vdots & \vdots & \ddots & \vdots \\ x(n-\tau-1) & x(n-\tau) & \cdots & x(n-2) \\ x(n-\tau) & x(n-\tau+1) & \cdots & x(n-1) \end{bmatrix} \tag{30}$$

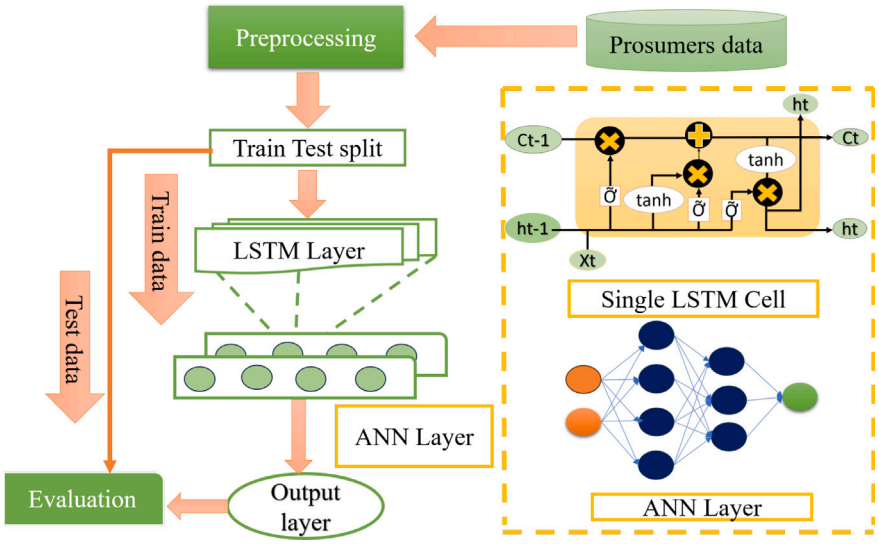


Fig. 5. A schematic illustration for compensating corrupted data by proposed LSTM-ANN.

Table 9
Predicting accuracy of the LSTM-ANN model.

Prosumer's identity	MAE	RMSE
5th	0.13	0.309
9th	0.196	0.278
11th	0.055	0.083
13th	0.035	0.043
22nd	0.077	0.166
26th	0.035	0.049

Table 10
Hyperparameters of the designed LSTM-ANN.

Hyperparameters	Value
LSTM Layer	Two layers; 256 and 128 neurons
Activation function (LSTM)	*tanh* (both LSTM layer)
ANN Layer (dense)	Two layers; 100 and 50 units
Activation function (ANN)	ReLU
Optimizer	Adam
Batch size	32
Epoch	100

$$Y = \begin{bmatrix} y(\tau+1) \\ y(\tau+2) \\ \vdots \\ y(n-1) \\ y(n) \end{bmatrix} \tag{31}$$

where the $x(1), x(2), \ldots, x(t)$ is the past input of any prosumer's energy trading data and the corresponding output will be the next samples of the data, i.e, $y(\tau+1)$.

Table 9 shows the evaluation metrics of LSTM-ANN for attack impact mitigation purpose of the victim prosumers and Table 10 summarizes the hyperparameters used for the LSTM-ANN model training purpose.

It is important to point out that the models have been implemented with a uniform architecture for all prosumers, with no individual customization, in this case. The LSTM-ANN hyperparameters are chosen by manually through a process of trial and error optimization, and the resulting model is regarded as a general model applicable to all victim of prosumers. The process of fine-tuning the hyperparameters can be expanded to include sophisticated search strategies or optimization algorithms in the future. Furthermore, in the future, client-specific hyperparameter adjustment can also be investigated. With Adam optimizers, the batch size has been set at 32 and 100 epochs. When training the model, these hyperparameters are kept the same for all of the data sets. The dataset is divided into training and testing sets in order to assess the model's performance on previously unexplored data. 25% of the data are put aside for testing, while the remaining 75% are used for training. However, the consistency of model architecture shows a minor edge over others in a constrained range. The general effectiveness of the forecasting models continues to be excellent, and this is emphasized throughout the results section through both the graphical and numerical result analyses. This decision to keep identical model architectures for all consumers enables a fair and unbiased evaluation of each prosumer's predicting skills.

Table 11 represents the comparison of attack impact mitigation accuracy for the two above-mentioned AI-based algorithms: LSTM, and LSTM-ANN. While the designed LSTM alone predicts more accurately for the 9th prosumer, LSTM-ANN outperforms LSTM when taking into account the performance of all six victim prosumers. Hence, this work chooses LSTM-ANN for data retrieving purpose after detecting any compromised data.

5. Performance assessment

This section represents the simulation results and analyzes the performance of the proposed security scheme. The decentralized energy transaction platform is assumed to consist of 30 prosumers, and its 20%, i.e., 6 prosumers, chosen randomly, are considered under different security breaches. Table 2 summarizes the attack strategy, followed by the prosumers' identities and the days on which the attacks take place. With the implementation of all the attack templates, the total 21% data of each prosumer is compromised.

5.1. Device configuration:

As most of the literature dealing with cyber security context of smart grid are limited to theoretical aspect only, this work gives an overall idea about the computational cost of the proposed framework for real-life implementation. The Jupyter Notebook environment (version 6.4.12) is used to run the Python code in order to wrap up the DT-based categorization problem. The computer has a 64-bit version of Windows 10 Home Single Language, 4 GB of RAM, and an Intel(R) Core(TM) $i5-8250U$ processor with a clock speed of 1.60–1.80 GHz.

Table 11
Compared predicting accuracy of LSTM and LSTM-ANN.

Prosumer's Identity	MAE of LSTM	MAE of LSTM-ANN	RMSE of LSTM	RMSE of LSTM-ANN
5th	0.375	0.13	0.664	0.309
9th	0.066	0.196	0.049	0.278
11th	0.095	0.055	0.165	0.083
13th	0.035	0.032	0.043	0.042
22nd	0.107	0.077	0.198	0.166
26th	0.052	0.035	0.074	0.049
Average	0.122	**0.088**	0.199	**0.155**

Table 12
Attack identification accuracy.

Prosumer's Identity	Attack identification accuracy (%)	Average identification accuracy (%)
5th	97.79	97.90
9th	98.74	
11th	96.21	
13th	98.11	
22nd	97.48	
26th	99.05	

Google Colab is used to run the code for creating the DL-based threat mitigation approach. The model's framework is created using Keras API and TensorFlow. The detection model can be trained instantly and can identify any compromised data immediately. Meanwhile, the forecasting model takes approximately 43 seconds to train, and about 0.85 s is required for forecasting purposes. In order to cope with the updated attack dynamics, the model can be trained with new data along with the previous one. Moreover, AI-specific chips can be utilized to reduce the burden on the system.

5.2. Analysis of evaluation metrics:

To justify the effectiveness of the proposed energy transaction security scheme, the following evaluation metrics are used:

5.2.1. Machine learning (ML)-based attack classifier

Accuracy:

A multiclass classification model's performance is frequently assessed using the algorithm's accuracy and the confusion matrix. The total number of correct predictions α^{cp} and the total number of predictions α are compared to determine accuracy $\tilde{\alpha}$, as defined in (32), to measure how accurate the trained model is.

$$\tilde{\alpha} = \frac{\alpha^{cp}}{\alpha} \tag{32}$$

where $\tilde{\alpha}$ is expressed in percentage. For example, if the confusion matrix, showing the trained model's performance for 5th prosumer, as per Fig. 6, is taken under consideration, then α^{cp} and α are 310, 317, respectively, and hence, $\tilde{\alpha}$ is 97.79%, which is summarized in Table 12. This table also enlists the accuracy of the DT model for identifying the attacks. Attack identification accuracies for prosumer 5th, 9th, 11th, 13th, 22nd, and 26th are 97.79%, 98.74%, 96.21%, 98.11%, 97.48%, and 99.05%, respectively.

Confusion Matrix:

Examining the confusion matrix is a significantly more effective technique to judge a classifier's performance. It is a tabular summary of how well a classification model performed on a set of test data. Figs. 6, 7, 8, 9, 10 and 11 represent the confusion matrix showing the attack detection performance of the designed DT for the 5th, 9th, 11th, 13th, 22nd and 26th prosumer, respectively.

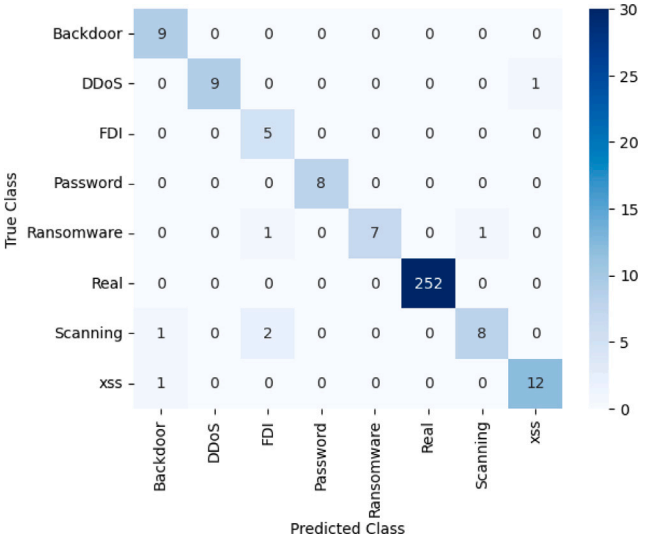


Fig. 6. Confusion matrix for 5th prosumer.

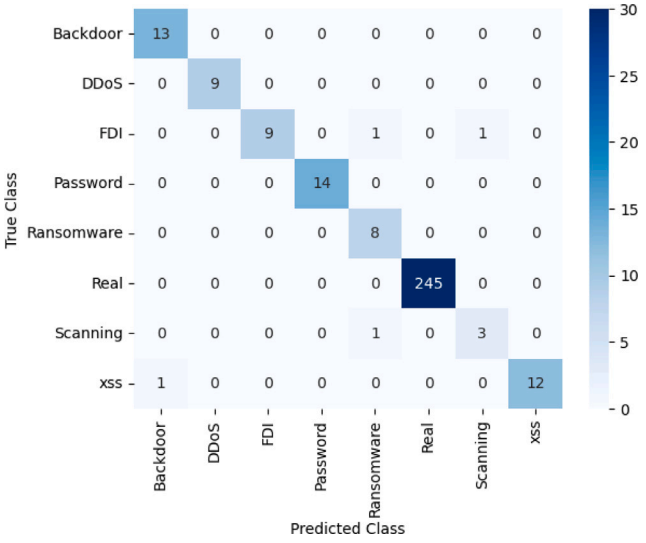


Fig. 7. Confusion matrix for 9th prosumer.

5.2.2. Deep learning (DL)-based attack impact mitigation model

Mean Absolute Error (MAE):

For DL-based issues related to regression, it is a frequently used loss function. The average absolute difference between the expected and actual values is measured by MAE. The performance of the developed model is frequently evaluated in the context of DL, using MAE as an assessment metric. If there are m samples in the dataset; $\hat{x_j}$ and x_j represent the expected and corresponding actual values under the jth

Fig. 8. Confusion matrix for 11th prosumer.

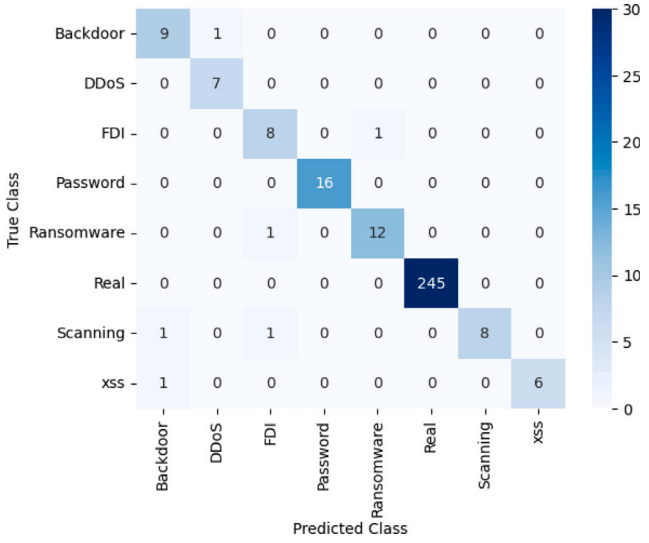


Fig. 9. Confusion matrix for 13th prosumer.

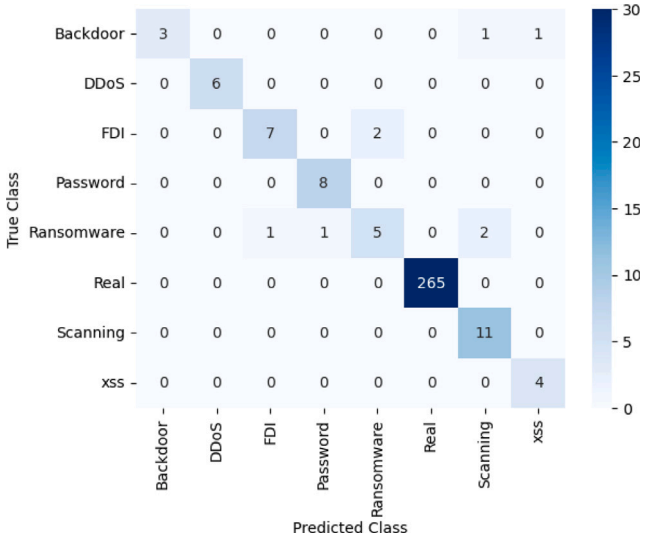


Fig. 10. Confusion matrix for 22nd prosumer.

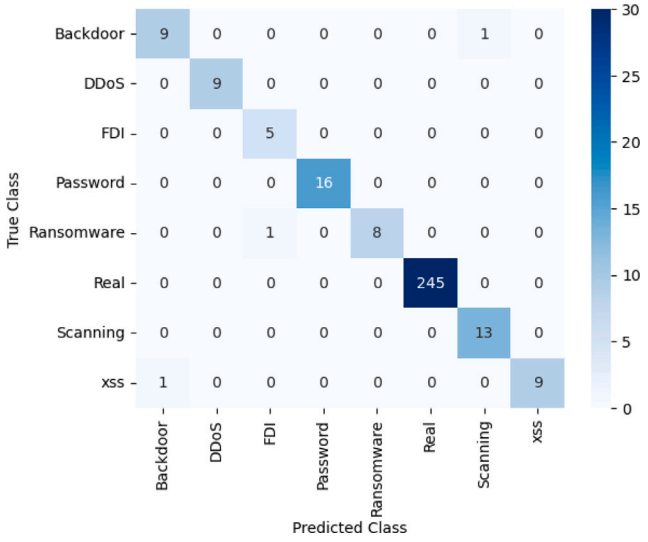


Fig. 11. Confusion matrix for 26th prosumer.

Table 13
Predicting accuracy of the model.

Prosumer's identity	MAE	RMSE
5th	0.13	0.309
9th	0.196	0.278
11th	0.055	0.083
13th	0.032	0.042
22nd	0.077	0.166
26th	0.035	0.049

independent variable, respectively, then conceptually, MAE may be expressed as:

$$MAE = \frac{1}{m}\sum_{j=1}^{m}|\hat{x}_j - x_j| \tag{33}$$

Root Mean Square Error (RMSE):

It is yet another typical assessment measure and loss function for DL-based regression problems. It calculates the square root of the average of the squared deviations between the expected and actual values. The accuracy and effectiveness of regression models are frequently evaluated using the RMSE. Mathematically:

$$RMSE = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\hat{x}_j - x_j)^2} \tag{34}$$

Symmetric Mean Absolute Percentage Error (SMAPE):

When dealing with little or zero actual values, it is a metric used to assess a forecasting model's performance based on proportional or relative errors. SMAPE evaluates the scale of the numbers along with the absolute percent difference between the predicted and actual values. It considers both positive and negative issues equally because it is symmetrical. The following is the formula for calculating SMAPE:

$$SMAPE = \frac{1}{m}\sum_{j=1}^{m}\frac{|a_j - f_j|}{(|a_j| + |f_j|)/2} \times 100 \tag{35}$$

where the true value at the index point j is a_j. The predicted value for index j is f_j and the total number of observations is m. The two evaluation metrics, MAE and RMSE, for each of the prosumers are listed in Table 13.

The forecasting model shows outstanding performance for all the prosumers, and its performances for the 5th, 9th, and 11th prosumers are shown in Figs. 12–14, respectively. Time series forecasting models use sequential data from the past to predict values for the future. The

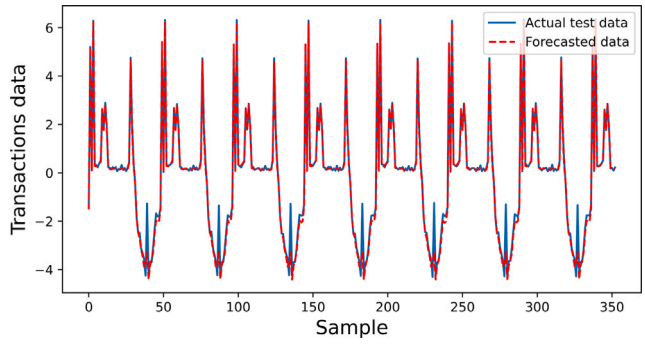


Fig. 12. Actual and predicted trading data for 5th prosumer.

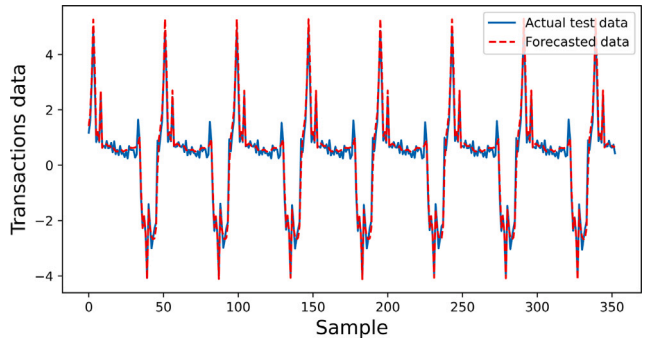


Fig. 13. Actual and predicted trading data for 9th prosumer.

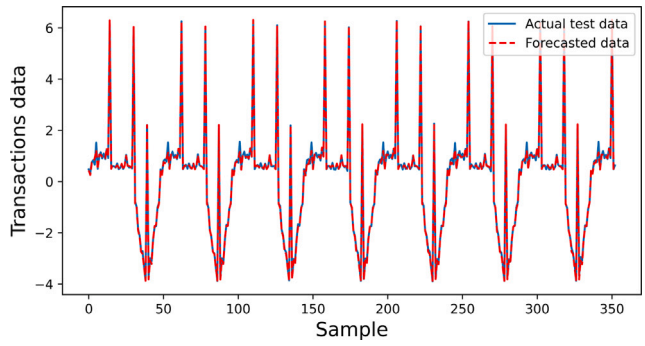


Fig. 14. Actual and predicted trading data for 11th prosumer.

designed model will predict the 31st data if the 1st through 30th data are provided as input. Now, if 32nd data is to be predicted then the model will take 2nd to 31st data as input and among these data the 31st is the previously predicted data. For some data points, there is a small difference between the actual and predicted data, which is displayed in Figs. 12–14, because the forecasted data is used for the subsequent forecast. 25% of the actual trading data of each prosumer is considered to evaluate the performances of the forecasting models by the above-mentioned two statistical error metrics. Both from the numerical and graphical point of view, it is evident that the proposed model can effectively mitigate attack impacts.

5.2.3. Performance evaluation to ensure data security

Finding the compromised data is the work's primary contribution. If the compromised data is accurately identified, the forecasting model will mitigate it. Upholding a secure and dependable information system requires ensuring data integrity, availability, and confidentiality. The effectiveness of the proposed model is demonstrated by the performance evaluation metrics listed below when dealing with data security.

Table 14
Evaluation metrics for data integrity.

Prosumer's identity	Precision (%)	Recall (%)	F1 Score (%)
5th	98.19	97.79	97.82
9th	98.89	98.74	98.74
11th	96.60	96.21	96.29
13th	98.27	98.11	98.12
22nd	97.56	97.48	97.33
26th	99.13	99.05	99.06

Performance Evaluation for Data Integrity:

Accuracy: The percentage of correctly classified instances relative to all instances is known as accuracy. Section 5.2.1 provides information on the decision tree model's accuracy in classifying instances of compromised data, and Table 12 lists the model's accuracy for all the victim prosumers.

Precision and Recall: precision is the ratio of true positives to all predicted positives, whereas Recall is the ratio of true positives to all actual positives. If TP, True Positive, represents the instances correctly predicted as positive; FP, False Positive, is the instances incorrectly predicted as positive, and FN, False Negative, indicates the instances incorrectly predicted as negative, then precision, recall, and F1 score can be represented as:

$$x_{Prcn} = \frac{TP}{TP + FP} \tag{36}$$

$$x_{Rcl} = \frac{TP}{TP + FN} \tag{37}$$

F1 Score: An impartial assessment of the model's performance in terms of recall and precision is given by the F1 score.

$$x_{F1\ Scr} = 2 \times \frac{x_{Prcn} \times x_{Rcl}}{x_{Prcn} + x_{Rcl}} \tag{38}$$

The corresponding F1 score, precision, and recall for the 5th prosumer are 97.82%, 98.19%, and 97.79%, respectively. These metrics are available for other prosumers in Table 14. These high evaluation metrics values suggest that the model is suitable for the task of identifying attacks because it is good at accurately classifying instances and finds a good balance between recall and precision.

Performance Evaluation for Data Unavailability:

False Positive Rate (FPR): It determines the percentage of events that the detection model flags as data unavailability attacks that are actually legitimate requests for data access. Better accuracy in differentiating between data unavailability attacks and legitimate data access is indicated by a lower FPR. It can be represented as:

$$x_{Fpr} = \frac{FP}{FP + TN} \tag{39}$$

where, TN, True Negative, represents the instances correctly predicted as negative.

False Negative Rate (FNR): It finds out what percentage of valid requests for data access are misclassified by the model as unavailability attacks. Better sensitivity in identifying unavailability attacks without misclassifying legitimate data access is indicated by a lower FNR. Mathematically, it is represented as:

$$x_{Fnr} = \frac{FN}{FN + TP} \tag{40}$$

Table 15 enlists the values of FPR and FNR for all the victim prosumers.

Predictive Accuracy: Predictive accuracy, commonly referred to as accuracy, measures how accurate a classification model is overall based on its predictions made on a dataset. The accuracy of the model is given in Table 12.

Performance Evaluation for Data Confidentiality:

Sensitivity and Specificity: Sensitivity measures how well the model can identify positive instances among all real positive instances. On the other hand, specificity measures how well the model can identify negative instances among all real negative instances. A higher value of them represents the detection model is performing well. Table 16 summarizes the sensitivity and specificity in % for all the victim prosumers.

Table 15
Evaluation metrics for data unavailability.

Prosumer's identity	FPR (%)	FNR (%)
5th	0.00	18.18
9th	0.00	7.14
11th	0.99	7.14
13th	0.33	18.18
22nd	0.64	0.00
26th	0.33	10.00

Table 16
Evaluation metrics for data confidentiality.

Prosumer's identity	Sensitivity (%)	Specificity (%)
5th	81.82	100.00
9th	92.86	100.00
11th	92.86	99.01
13th	81.82	99.67
22nd	100.00	99.36
26th	90.00	99.67

Table 17
Proposed LSTM-ANN's attack impact mitigation effectiveness.

Security breaches	SMAPE of corrupted trading data	SMAPE of predicted trading data by proposed LSTM-ANN
FD Injection	132.21%	7.816%
DDoS	199.982%	7.918%
Password	100.168%	7.446%
Backdoor	190%	7.311%
Ransomware	33.598%	8.186%
Scanning	50%	8.265%
XSs	195%	7.85%

5.3. Attack impact mitigation's efficiency: A case study

This section investigates how effective the LSTM-ANN-based model is in predicting original trading data under different security breaches for a particular prosumer, say 11th. The 3% data of the prosumer is considered corrupted by each category of the attack. Figs. 15(a)–15(g) help to analyze the original trading data, corrupted trading data, and mitigated trading data by the proposed LSTM-ANN scheme under DDoS, FD injection, password, BDoor, ransomware, Scan, and XSs, respectively. It is evident from all of the Figs that the trading data that has been mitigated is nearly linked to the original one, demonstrating the effectiveness of the suggested attack impact mitigation strategy. The numerical values of the SMAPE analysis of the prosumer for all the attacks are listed in Table 17. For example, the SMAPE of the compromised data having a DDoS attack is 199.982%, which is reduced to only 7.918% when the mitigated trading data is used. So, not only from a graphical point of view but also from a numeric value analysis, it appears that the proposed LSTM-ANN can effectively mitigate the impacts of different attacks on trading data (see Tables 14–16).

5.4. Real-world application

The initial stage in the proposed methodology is to identify any compromised data during energy trade. In case of detecting any corrupted data incidence, according to the framework, the compromised data will be exchanged by the predicted data. In order to justify the proposed architecture's effectiveness in real-world context, the attack identification model's performance is evaluated by applying the dataset given by an Australian network service provider [73], which is summarized in Table 18. The dataset contains three years of half-hour electricity data for 300 prosumers. We have considered the same environment, i.e., the same energy transaction framework and attack strategy as mentioned previously, to evaluate the model's efficacy for the real-world data. For the victim prosumers, at first the net metering data is calculated by following equation (1). Then all of the attack variants are implemented by following Eqs. (5), (7), (8), (9), (10), (11) and (12).

The confusion matrices showing the attack detection performance of the designed DT for the 5th, 9th, 11th, 13th, 22nd and 26th prosumer, for the real-world data, are represented by Figs. 16, 17, 18, 19, 20 and 21 respectively. The analysis of confusion metrics for the real-world data reveals that the model can successfully identify the actual and the corrupted trading data. However, there is little conflict among the various attack categories and this reduces the identification accuracy of the model to some extent when compared to the synthetic data context. Table 19 shows the model's performance comparison for synthetic data and real-world data. From the evaluation metrics' comparison of synthetic data with the real-world data, it can be said that the synthetic dataset is closely related to the real-world condition.

While the suggested model exhibits promising performance in controlled circumstances, its actual implementation may need significant computational resources and real-time data processing capabilities, which may cause little issues in resource-constrained or high-availability systems.

6. Conclusion

The implications of successful cyber assaults on IoT devices from the standpoint of the prosumer have not received enough attention, despite the fact that attack surfaces, threat vectors, and vulnerabilities in IoT devices are the subject of many studies. By examining both the graphical and numerical representations of the energy transaction data, this research has bridged this gap, and the usefulness of the proposed approach has been demonstrated in this paper. Improving the security of energy trading within a network of prosumers has been focused on in this paper by developing a comprehensive framework for not only detecting and preventing a specific prosumer from reacting to a compromised trading value but also mitigating different attacks' effects by replacing this trade value with the forecasted one. The altered values have been saved for further analysis in the central entity in addition to the detection, prevention, and mitigation functions. This storage can serve as a significant resource for understanding the nature of attacks, spotting patterns, and implementing improved security measures in the future.

In this paper, a supervised ML approach based on DT has been used to identify several sorts of assaults on data availability, secrecy, and integrity. Six prosumers — the 5th, 9th, 11th, 13th, 22nd and 26th — have been selected at random under various security breaches to test the effectiveness of the proposed ML-based attack detection scheme. For each of the corresponding prosumers, the attack identification accuracy of the suggested technique has been 97.79%, 98.74%, 96.21%, 98.11%, 97.48%, and 99.05%. Additionally, the confusion matrices have illustrated the model's effectiveness graphically. The integrity and dependability of the energy trading process have been ensured upon discovering an attack because the compromised data has been filtered out and replaced with a predicted trade value. The combination of LSTM and ANN in the proposed model has contributed to accurate predictions. With the proposed method, past data can effectively be analyzed in order to forecast future energy trading trends. The use of the suggested LSTM-ANN model, as shown by graphical and numerical result analysis, can aid in making informed decisions and reduce potential hazards related to damaged or compromised data. The SMAPE analysis has also been conducted for a certain prosumer, i.e., 11th, taking into account both the defective data and the mitigated data. A closer look at the SMAPE numerical values can reveal the close connection between the original trade data and the mitigated trading data, and the dissimilarity between corrupted data and original data.

Thus, this work has emphasized the potentiality of ML-based methods for protecting energy trading networks supported by the efficiency of proactive detection, filtering, and prediction strategies in preserving the integrity, confidentiality, and availability of energy trading among

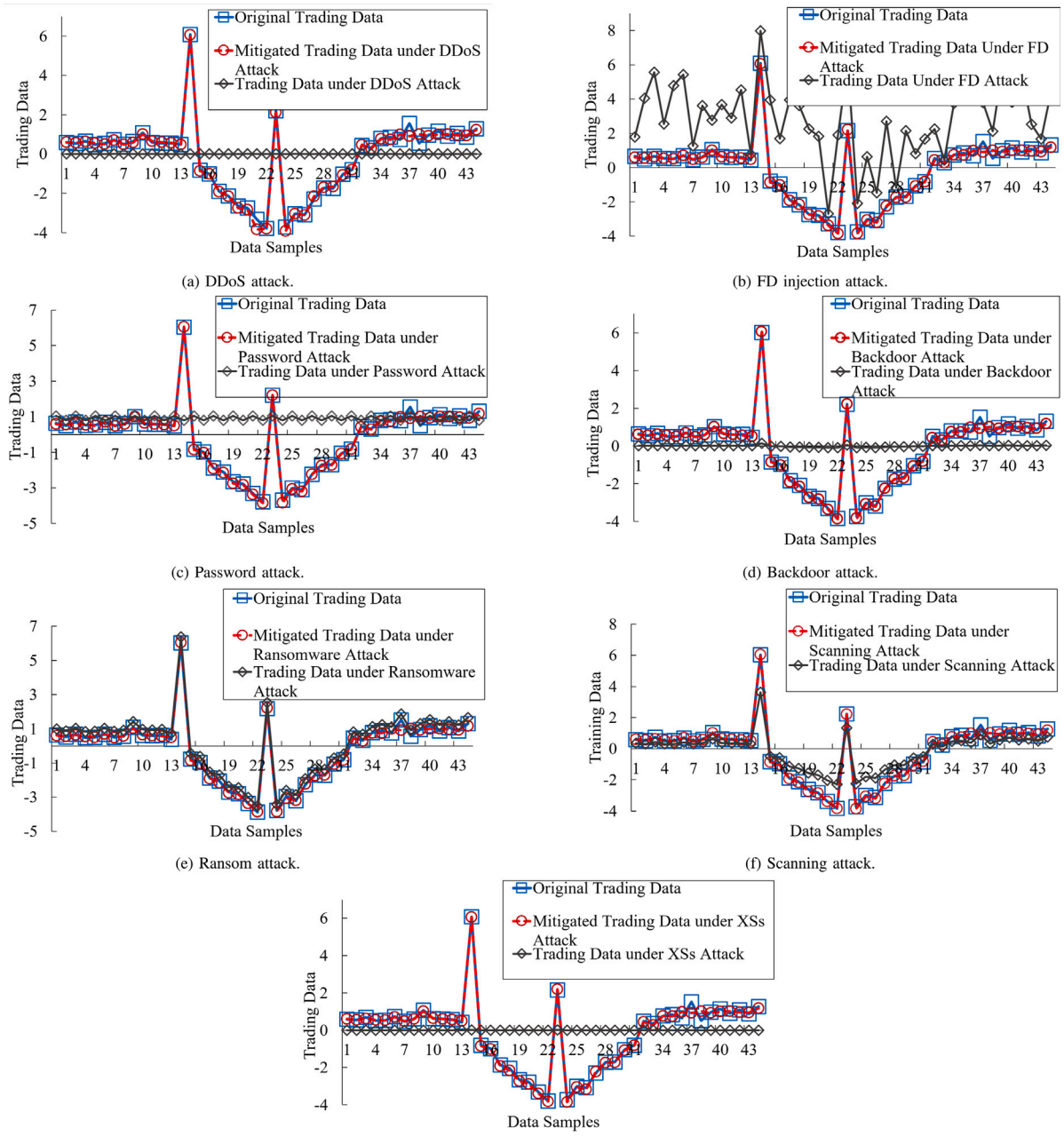


Fig. 15. Proposed LSTM-ANN's performance in predicting the original trading data of 11th prosumer during different attacks.

Table 18
Evaluation metrics of DT for real-world data [73].

Prosumer's identity	Identification accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
5th	95.90	96.33	95.90	95.98
9th	94.01	94.26	94.01	94.06
11th	94.32	95.38	94.32	94.53
13th	94.01	94.58	94.01	93.86
22nd	95.58	96.06	95.58	95.70
26th	95.27	95.63	95.27	95.35

prosumers by merging DT and LSTM-ANN architectures. This research can offer important insights for guaranteeing safe and effective energy transactions through IoT-based networks of prosumers as the energy sector continues to develop. Future research can be undertaken to determine how the mitigated trading values would affect the financial aspects and network deployment of the LEM.

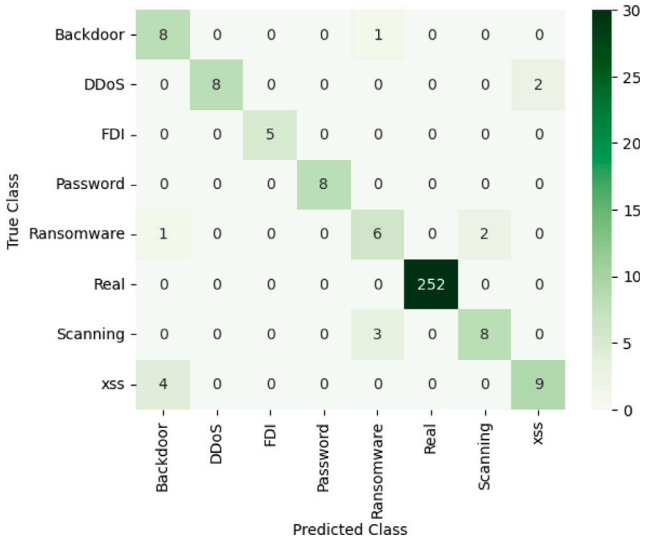


Fig. 16. Confusion matrix of 5th prosumer for real data.

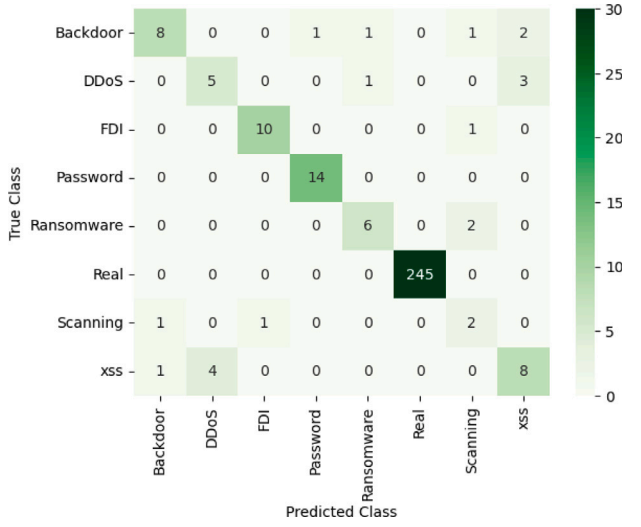


Fig. 17. Confusion matrix of 9th prosumer for real data.

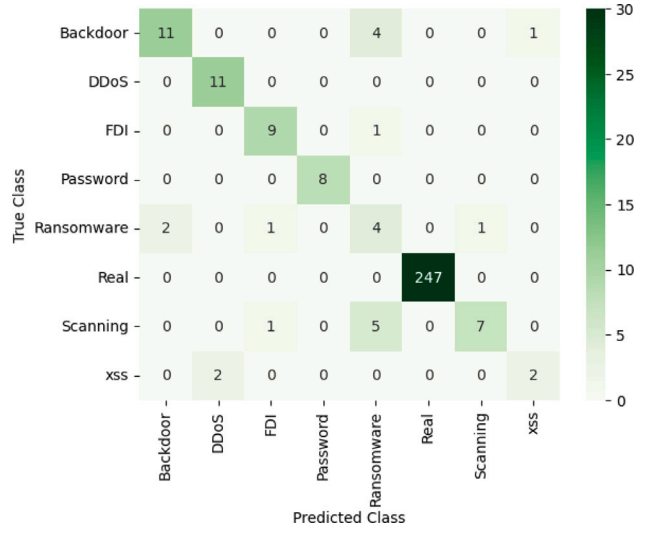


Fig. 18. Confusion matrix of 11th prosumer for real data.

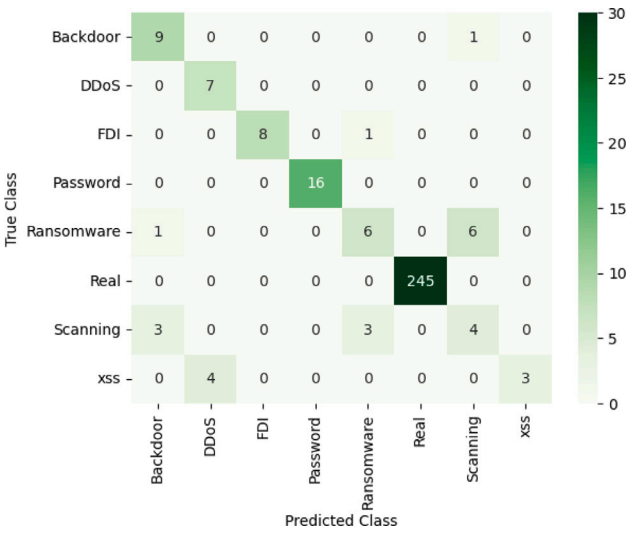


Fig. 19. Confusion matrix of 13th prosumer for real data.

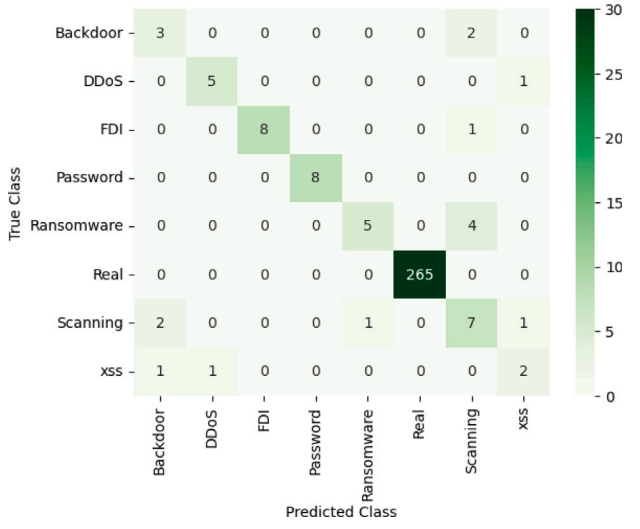


Fig. 20. Confusion matrix of 22nd prosumer for real data.

Fig. 21. Confusion matrix of 26th prosumer for real data.

Table 19
Compared identification accuracy of DT for synthetic and real-world data.

Prosumer's identity	Identification accuracy for synthetic data (%)	Identification accuracy for real-world data (%)
5th	97.79	95.90
9th	98.74	94.01
11th	96.21	94.32
13th	98.11	94.01
22nd	97.48	95.58
26th	99.05	95.27
Average accuracy (%)	97.90	94.85

CRediT authorship contribution statement

Fariya Tabassum: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Md. Rashidul Islam:** Writing – review & editing, Methodology, Formal analysis, Conceptualization. **M. Imran Azim:** Writing – review & editing, Supervision, Investigation, Conceptualization. **M.A. Rahman:** Validation, Software. **Md. Omer Faruque:** Validation, Software. **Sk.A. Shezan:** Writing – review & editing, Supervision. **M.J. Hossain:** Writing – review & editing, 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] M.I. Azim, M.T. Islam, J.H. Rakib, M.R. Islam, L. Ali, S. Alzahrani, H. Masrur, S. Muyeen, Coalition game theoretic P2P trading in a distribution network integrity-ensured local energy market, Sustain. Energy Grids Netw. 36 (2023) 101186.

[2] A.V. Christopher, D. Samiappan, R. Rengaswamy, Automatic adaptive synchronization (A2S): A demand-based automatic synchronization for distribution generators in islanding mode, Knowl.-Based Syst. 275 (2023) 110641.

[3] P. Grammatikos, M. Paolone, J.-Y. Le Boudec, Design of cost functions for the real-time control of microgrids hosting distributed energy-storage systems, Sustain. Energy Grids Netw. 35 (2023) 101141.

[4] D. Liu, P. Cheng, J. Cheng, J. Liu, M. Lu, F. Jiang, Improved reinforcement learning-based real-time energy scheduling for prosumer with elastic loads in smart grid, Knowl.-Based Syst. 280 (2023) 111004.

[5] S. Hussain, M.I. Azim, C. Lai, U. Eicker, New coordination framework for smart home peer-to-peer trading to reduce impact on distribution transformer, Energy 284 (2023) 129297.

[6] M.I. Azim, W. Tushar, T.K. Saha, C. Yuen, D. Smith, Peer-to-peer kilowatt and negawatt trading: A review of challenges and recent advances in distribution networks, Renew. Sustain. Energy Rev. 169 (2022) 112908.

[7] S. Hussain, M.I. Azim, C. Lai, U. Eicker, Multi-stage optimization for energy management and trading for smart homes considering operational constraints of a distribution network, Energy Build. (2023) 113722.

[8] A. Malkawi, S. Ervin, X. Han, E.X. Chen, S. Lim, S. Ampanavos, P. Howard, Design and applications of an IoT architecture for data-driven smart building operations and experimentation, Energy Build. (2023) 113291.

[9] R.-V. Tkachuk, D. Ilie, R. Robert, V. Kebande, K. Tutschku, Towards efficient privacy and trust in decentralized blockchain-based peer-to-peer renewable energy marketplace, Sustain. Energy Grids Netw. 35 (2023) 101146.

[10] M.H.P. Rizi, S.A.H. Seno, A systematic review of technologies and solutions to improve security and privacy protection of citizens in the smart city, Internet of Things 20 (2022) 100584, 1–33.

[11] S. Ahmadzadeh, G. Parr, W. Zhao, A review on communication aspects of demand response management for future 5G IoT-based smart grids, IEEE Access 9 (2021) 77555–77571.

[12] R.R. Mohassel, A. Fung, F. Mohammadi, K. Raahemifar, A survey on advanced metering infrastructure, Int. J. Electr. Power Energy Syst. 63 (2014) 473–484.

[13] R. Qi, Q. Li, Z. Luo, J. Zheng, S. Shao, Deep semi-supervised electricity theft detection in AMI for sustainable and secure smart grids, Sustain. Energy Grids Netw. 36 (2023) 101219.

[14] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, T. Melodia, Machine learning for wireless communications in the internet of things: A comprehensive survey, Ad Hoc Netw. 93 (2019) 101913.

[15] P. Zhuang, T. Zamir, H. Liang, Blockchain for cybersecurity in smart grid: A comprehensive survey, IEEE Trans. Ind. Inform. 17 (1) (2020) 3–19.

[16] M. Mohammadpourfard, Y. Weng, M. Pechenizkiy, M. Tajdinian, B. Mohammadi-Ivatloo, Ensuring cybersecurity of smart grid against data integrity attacks under concept drift, Int. J. Electr. Power Energy Syst. 119 (2020) 105947.

[17] S. Zhang, T. Zheng, B. Wang, A privacy protection scheme for smart meter that can verify terminal's trustworthiness, Int. J. Electr. Power Energy Syst. 108 (2019) 117–124.

[18] H. Zhang, B. Liu, H. Wu, Smart grid cyber–physical attack and defense: A review, IEEE Access 9 (2021) 29641–29659.

[19] A. Huseinović, S. Mrdović, K. Bicakci, S. Uludag, A survey of denial-of-service attacks and solutions in the smart grid, IEEE Access 8 (2020) 177447–177470.

[20] M.Z. Gunduz, R. Das, Cyber-security on smart grid: Threats and potential solutions, Computer networks 169 (2020) 107094.

[21] P.A. Giglou, S.N. Ravadanegh, Defending against false data injection attack on demand response program: A bi-level strategy, Sustain. Energy Grids Netw. 27 (2021) 100506.

[22] S. Mohammadi, F. Eliassen, Y. Zhang, H.-A. Jacobsen, Detecting false data injection attacks in peer to peer energy trading using machine learning, IEEE Trans. Dependable Secure Comput. 19 (5) (2021) 3417–3431.

[23] C. Zhang, F. Luo, M. Sun, G. Ranzi, Modeling and defending advanced metering infrastructure subjected to distributed denial-of-service attacks, IEEE Trans. Netw. Sci. Eng. 8 (3) (2020) 2106–2117.

[24] S.Y. Diaba, M. Elmusrati, Proposed algorithm for smart grid ddos detection based on deep learning, Neural Netw. 159 (2023) 175–184.

[25] S. Wang, H. Zhang, S. Qin, W. Li, T. Tu, A. Shen, W. Liu, Krprotector: Detection and files protection for IoT devices on android without root against ransomware based on decoys, IEEE Internet Things J. 9 (19) (2022) 18251–18266.

[26] A. Guan, C.-M. Chen, A novel verification scheme to resist online password guessing attacks, IEEE Trans. Dependable Secure Comput. 19 (6) (2022) 4285–4293.

[27] C. Birkinshaw, E. Rouka, V.G. Vassilakis, Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks, J. Netw. Comput. Appl. 136 (2019) 71–85.

[28] V. Visoottiviseth, P. Sakarin, J. Thongwilai, T. Choobanjong, Signature-based and behavior-based attack detection with machine learning for home iot devices, in: 2020 IEEE REGION 10 CONFERENCE, TENCON, IEEE, 2020, pp. 829–834.

[29] F. Qi, Y. Chen, M. Li, Y. Yao, Z. Liu, M. Sun, Onion: A simple and effective defense against textual backdoor attacks, 2020, arXiv preprint arXiv:2011.10369.

[30] J.W. Goodell, S. Kumar, W.M. Lim, D. Pattnaik, Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis, J. Behav. Exp. Finance 32 (2021) 100577.

[31] L. Chen, P. Chen, Z. Lin, Artificial intelligence in education: A review, IEEE Access 8 (2020) 75264–75278.

[32] A.İ. Tekkeşin others, Artificial intelligence in healthcare: past, present and future, Anatol. J. Cardiol. 22 (Suppl 2) (2019) 8–9.

[33] G. Zeba, M. Dabić, M. Čičak, T. Daim, H. Yalcin, Technology mining: Artificial intelligence in manufacturing, Technol. Forecast. Soc. Change 171 (2021) 120971.

[34] K. Shaukat, S. Luo, V. Varadharajan, I.A. Hameed, M. Xu, A survey on machine learning techniques for cyber security in the last decade, IEEE Access 8 (2020) 222310–222354.

[35] S.D. Milić, Ž. DJurović, M.D. Stojanović, Data science and machine learning in the IIOT concepts of power plants, Int. J. Electr. Power Energy Syst. 145 (2023) 108711.

[36] L. Ali, M.I. Azim, N.B. Ojha, J. Peters, V. Bhandari, A. Menon, J. Green, S. Muyeen, Integrating forecasting service and Gen2 blockchain into a local energy trading platform to promote sustainability goals, IEEE Access 12 (2023) 2941–2964.

[37] S. Rizvi, A. Kurtz, J. Pfeffer, M. Rizvi, Securing the internet of things (IOT): A security taxonomy for iot, in: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), IEEE, 2018, pp. 163–168.

[38] B. Hou, J. Gao, X. Guo, T. Baker, Y. Zhang, Y. Wen, Z. Liu, Mitigating the backdoor attack by federated filters for industrial IoT applications, IEEE Trans. Ind. Inform. 18 (5) (2021) 3562–3571.

[39] M. Cui, J. Wang, M. Yue, Machine learning-based anomaly detection for load forecasting under cyberattacks, IEEE Trans. Smart Grid 10 (5) (2019) 5724–5734.

[40] R. Leszczyna, A review of standards with cybersecurity requirements for smart grid, Comput. Secur. 77 (2018) 262–276.

[41] M. Almseidin, M. Al-Kasassbeh, S. Kovacs, Detecting slow port scan using fuzzy rule interpolation, in: 2019 2nd International Conference on New Trends in Computing Sciences, ICTCS, IEEE, 2019, 1–6.

[42] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, N. Ghani, Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2702–2733.
[43] P. Chaudhary, B.B. Gupta, K.T. Chui, S. Yamaguchi, Shielding smart home IOT devices against adverse effects of XSS using AI model, in: 2021 IEEE International Conference on Consumer Electronics, ICCE, IEEE, 2021, p. 2021.
[44] N.M. Karie, N.M. Sahri, W. Yang, C. Valli, V.R. Kebande, A review of security standards and frameworks for IoT-based smart environments, IEEE Access 9 (2021) 121975–121995.
[45] A. Khurshid, R. Alsaaidi, M. Aslam, S. Raza, Eu cybersecurity act and iot certification: landscape, perspective and a proposed template scheme, IEEE Access 10 (2022) 129932–129948.
[46] Y. Liu, Z. Li, M. Backes, Y. Shen, Y. Zhang, Backdoor attacks against dataset distillation, 2023, arXiv preprint arXiv:2301.01197.
[47] Y. Liu, X. Ma, J. Bailey, F. Lu, Reflection backdoor: A natural backdoor attack on deep neural networks, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August (2020) 23–28, Proceedings, Part X 16, Springer, 2020, pp. 182–199.
[48] K. Sonar, H. Upadhyay, A survey: Ddos attack on internet of things, Int. J. Eng. Res. Dev. 10 (11) (2014) 58–63.
[49] A.S. Yahaya, N. Javaid, F.A. Alzahrani, A. Rehman, I. Ullah, A. Shahid, M. Shafiq, Blockchain based sustainable local energy trading considering home energy management and demurrage mechanism, Sustainability 12 (8) (2020) 3385.
[50] D. Stiawan, M. Idris, R.F. Malik, S. Nurmaini, N. Alsharif, R. Budiarto others, Investigating brute force attack patterns in IoT network, J. Electr. Comput. Eng. (2019).
[51] J. Kalbantner, K. Markantonakis, D. Hurley-Smith, R.N. Akram, B. Semal, P2PEdge: A decentralised, scalable P2P architecture for energy trading in real-time, Energies 14 (3) (2021) 606.
[52] A.S. Tummala, R.K. Inapakurthi, A two-stage kalman filter for cyber-attack detection in automatic generation control system, J. Mod. Power Syst. Clean Energy 10 (1) (2021) 50–59.
[53] J.W. Goodell, S. Corbet, Commodity market exposure to energy-firm distress: Evidence from the colonial pipeline ransomware attack, Finance Res. Lett. 51 (2023) 103329.
[54] L. Sun, K. Zhou, X. Zhang, S. Yang, Outlier data treatment methods toward smart grid applications, IEEE Access 6 (2018) 39849–39859.
[55] W. Chen, K. Zhou, S. Yang, C. Wu, Data quality of electricity consumption data in a smart grid environment, Renew. Sustain. Energy Rev. 75 (2017) 98–105.
[56] L.M. Raggi, F.C. Trindade, V.C. Cunha, W. Freitas, Non-technical loss identification by using data analytics and customer smart meters, IEEE Trans. Power Deliv. 35 (6) (2020) 2700–2710.
[57] H. Wang, G. Li, Z. Wang, Fast SVM classifier for large-scale classification problems, Inform. Sci. 642 (2023) 119136.
[58] A. Delilbasic, B. Le Saux, M. Riedel, K. Michielsen, G. Cavallaro, A single-step multiclass SVM based on quantum annealing for remote sensing data classification, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. (2023).
[59] T.B. Krishna, P. Kokil, Automated classification of common maternal fetal ultrasound planes using multi-layer perceptron with deep feature integration, Biomed. Signal Process. Control 86 (2023) 105283.
[60] V. Kumar, S. Agrawal, A multi-layer Perceptron–Markov chain based lulc change analysis and prediction using remote sensing data in Prayagraj district, India, Environ. Monit. Assess. 195 (5) (2023) 619.
[61] B.A. de Abreu, G. Paim, M. Grellert, S. Bampi, C2pax: Complexity-aware constant parameter approximation for energy-efficient tree-based machine learning accelerators, IEEE Trans. Circuits Syst. I. Regul. Pap. 69 (7) (2022) 2683–2693.
[62] V.G. Costa, C.E. Pedreira, Recent advances in decision trees: An updated survey, Artif. Intell. Rev. 56 (5) (2023) 4765–4800.
[63] M. Bansal, A. Goyal, A. Choudhary, A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning, Decis. Anal. J. 3 (2022) 100071.
[64] Y. El Zein, M. Lemay, K. Huguenin, Privatree: Collaborative privacy-preserving training of decision trees on biomedical data, IEEE/ACM Trans. Comput. Biol. Bioinform. (2023).
[65] F.F. Fadoul, A.A. Hassan, R. Çağlar, Assessing the feasibility of integrating renewable energy: Decision tree analysis for parameter evaluation and LSTM forecasting for solar and wind power generation in a campus microgrid, IEEE Access (2023).
[66] B. Lindemann, B. Maschler, N. Sahlab, M. Weyrich, A survey on anomaly detection for technical systems using LSTM networks, Comput. Ind. 131 (2021) 103498.
[67] M. Abdel-Nasser, K. Mahmoud, M. Lehtonen, Reliable solar irradiance forecasting approach based on Choquet integral and deep LSTMs, IEEE Trans. Ind. Inform. 17 (3) (2020) 1873–1881.
[68] M. Kowsher, A. Tahabilder, M.Z.I. Sanjid, N.J. Prottasha, M.S. Uddin, M.A. Hossain, M.A.K. Jilani, LSTM-ANN & BiLSTM-ANN: Hybrid deep learning models for enhanced classification accuracy, Procedia Comput. Sci. 193 (2021) 131–140.
[69] W. Li, X. Wang, L. Wang, L. Jia, R. Song, Z. Fu, W. Xu, An LSTM and ANN fusion dynamic model of a proton exchange membrane fuel cell, IEEE Trans. Ind. Inform. 19 (4) (2022) 5743–5751.
[70] N.F. Ali, M. Atef, An efficient hybrid LSTM-ANN joint classification-regression model for PPG based blood pressure monitoring, Biomed. Signal Process. Control 84 (2023) 104782.
[71] Y. Hu, J. Ni, L. Wen, A hybrid deep learning approach by integrating LSTM-ANN networks with garch model for copper price volatility prediction, Phys. A 557 (2020) 124907.
[72] A. Bhatt, W. Ongsakul, J.G. Singh others, Sliding window approach with first-order differencing for very short-term solar irradiance forecasting using deep learning models, Sustain. Energy Technol. Assess. 50 (2022) 101864.
[73] Solar home electricity data, 2024, https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data [Accessed on May 15 2024].