Energy Reports 10 (2023) 3001–3019

Contents lists available at ScienceDirect

Energy Reports

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

Research paper

Step towards secure and reliable smart grids in Industry 5.0: A federated learning assisted hybrid deep learning model for electricity theft detection using smart meters

Muhammad Hamza Zafar [a,1], Syed Muhammad Salman Bukhari [b,1], Mohamad Abou Houran [c], Syed Kumayl Raza Moosavi [d], Majad Mansoor [e,f], Nedaa Al-Tawalbeh [g], Filippo Sanfilippo [a,h,*]

[a] Department of Engineering Sciences, University of Agder, Grimstad, 4879, Norway
[b] Department of Electrical Engineering, Capital University of Science and Technology, Islamabad, 44000, Pakistan
[c] School of Electrical Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
[d] School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, 44000, Pakistan
[e] Department of Automation, University of Science and Technology of China, 28796, China
[f] Ningbo China Institute for Supply Chain Innovation, Ningbo, 28796, China
[g] Department of Renewable Energy Engineering, Al al-Bayt University, Mafraq, 25113, Jordan
[h] Department of Software Engineering, Kaunas University of Technology, Kaunas, 51368, Lithuania

ARTICLE INFO

Keywords:
Smart meters
Theft detection
Deep learning
Federated learning
Security and reliability
Smart grid
Industry 5.0

ABSTRACT

The integration of Smart Grid technology and conceptual Industry 5.0 has paved the way for advanced energy management systems that enhance efficiency and revolutionized the parallel integration of power sources in a sustainable manner. However, this digitization has opened a new stream of the threat and opportunities of electricity theft posing a significant challenge to the security and reliability of Smart Grid networks. In this paper, we propose a secure and reliable theft detection technique using deep federated learning (FL) mechanism. The technique leverages the collaborative power of FL to train a Convolutional Gated Recurrent Unit (ConvGRU) model on distributed data sources without compromising data privacy. The training deep learning model backbone consists of a ConvGRU model that combines convolutional and gated recurrent units to capture spatial and temporal patterns in electricity consumption data. An improvised preprocessing mechanism and hyperparameter tuning is done to facilitate FL mechanism. The halving randomized search algorithm is used for hyperparameters tuning of the ConvGRU model. The impact of hyperparameters involved in the ConvGRU model such as number of layers, filters, kernel size, activation function, pooling, GRU layers, hidden state dimension, learning rate, and the dropout rate is elaborated. The proposed technique achieves promising results, with high accuracy, precision, recall, and F1 score, demonstrating its efficacy in detecting electricity theft in Smart Grid networks. Comparative analysis with existing techniques reveal the superior performance of the deep FL-based ConvGRU model. The findings highlight the potential of this approach in enhancing the security and efficiency of Smart Grid systems while preserving data privacy.

1. Introduction

A smart grid is comprised of intelligent sensors and meters, which establish connections to central servers or cloud platforms via either wireless or wired networks. A smart grid may manage electrical energy more effectively than a traditional system (Gul et al., 2020; Bohani et al., 2021; Mujeeb and Javaid, 2019). Framework analysis and dynamic load scheduling are employed in smart grids to achieve effective use of electricity (Marzband et al., 2018; Jadidbonab et al., 2020). For example, a hierarchical energy management system is presented in Gholinejad et al. (2020) with the goal of minimizing peak hours and selling more electricity for less money. In the work by Mian Qaisar (2020), a strategy rooted in information gap decision theory is introduced with the aim of mitigating the impact of the unpredictable characteristics inherent in renewable energy sources. Efficient energy resource use is vital for sustainable social and economic development due to rising energy costs and scarcity. Smart grids proved to be more efficient in operation and allow active monitoring of data hence

[*] Corresponding author at: Department of Engineering Sciences, University of Agder, Grimstad, 4879, Norway.
E-mail address: filippo.sanfilippo@uia.no (F. Sanfilippo).
[1] The authors contributed equally to this work.

https://doi.org/10.1016/j.egyr.2023.09.100
Received 3 August 2023; Received in revised form 13 September 2023; Accepted 18 September 2023
Available online 29 September 2023

are an essential component of future power grid infrastructure. The Smart Grid system seamlessly combines power system architecture and advanced computer technology to holistically oversee and control energy consumption on a comprehensive scale (Khan et al., 2020). This intelligent system monitors the usage patterns and behavior of customers connected to the system (Hasan et al., 2019) enabling customers and utility providers with regulation and forecast capability by fusing modern digital technologies with the existing electrical infrastructure. These concepts lead to the notion of the Energy Internet (EI). A key component of the EI is the bidirectional interchange of information and energy Cao et al. (2018), Wang (2003). As outlined in both Karnouskos et al. (2007) and Jiang et al. (2014), the Advanced Metering Infrastructure (AMI) serves as the fundamental basis for the EI. The AMI gives the power utilities highly detailed information about energy use. This is accomplished by strategically deploying smart meters for accurate load forecasting (Zheng et al., 2018) modeling of user consumption behavior (Wang et al., 2016) and demand response (Sun et al., 2018). Technical and nontechnical losses can occur during the transmission and distribution of power. Power transmission and distribution losses arise due to technical factors and fall within the scope of regulation (Henriques et al., 2020). Conversely, non-technical losses (NTL), encompassing issues like power theft, unethical conduct by utility personnel, and irregular billing, represent the primary sources of such losses (Savian et al., 2021). It has been estimated in Hussain et al. (2021) that the NTL costs utilities throughout the world 96 billion USD per year. Power companies, engineers, and researchers are working to decrease NTL utilizing a range of cutting-edge and effective methods because of the large economic loss (Arango et al., 2017).

Smart meter-based EI is highly effective against energy theft. Such a method might be utilized to immediately communicate the data to the utility while also remotely tracking consumer usage statistics and recording any suspicious activities. Smart meters provide a variety of advantages, but due to the high deployment and maintenance costs, they are not practical for nations that are facing severe economic difficulties. Before such devices may be extensively utilized, emerging cyber dangers must be appropriately handled. AMI's unique characteristics pose challenges in securing information flow within the EI. Smart meter data alteration by malicious users can lead to distinct power theft on the EI, differing from conventional grid tampering (Zheng et al., 2018). Artificial intelligence and deep learning algorithms are playing critical role in the energy sector for energy optimization (Khan et al., 2021; Zafar et al., 2023). Algorithms for artificial intelligence (AI) can automatically track users' patterns of energy use. Examining data collected by smart meters holds the potential to accurately detect instances of power theft. Cases of organized energy theft have been reported by renowned organizations including the US Federal Bureau of Investigation and the Fujian Daily (Zheng et al., 2018). These were the cause of a substantial NTL and were based on the use of tools and strategies against smart meters. In order to successfully combat the NTL issue, effective EI-based tactics for energy theft detection are required, as traditional detection techniques such as the use of technical employees or video surveillance are time-consuming and labor-intensive. Network measures cannot be tampered with by the intruders. Consequently, disparities may arise between the data from smart meters and the states of the system. Achieving a heightened degree of accuracy in detecting theft is feasible, albeit with the trade-off of introducing additional equipment. These strategies are impractical for many power providers due to rising maintenance and sensor deployment costs. Contrary to hardware-based solutions, non-hardware-based energy-theft detection methods do not call for additional NTL detecting apparatus. The two main categories of these techniques are those based on AI and those based on game theory (Jokar et al., 2015) The foundation of game theory-based methods for detecting the NTL is the creation of a game between the service provider and fraudulent customers. Although conventional prevention is less expensive initially but faces challenges in defining crucial responsibilities among participants, offenders, regulatory bodies, and distributors. AI-driven solutions provide a pragmatic approach by leveraging machine learning methodologies like classification and clustering. These techniques enable the identification of aberrant users through the analysis of consumer load profiles. It is believed that fraudulent users exhibit consumption patterns distinct from trustworthy clients.

The summary of literature for federated and non-federated learning-based electricity theft detection in smart grid is presented in Table 1.

This work is focused on methods for data analysis and data privacy protection. Different federated learning (FL)-based deep learning approaches are tried and evaluated using the data consumption energy of diverse customers to learn about and detect anomalous consumption behavior. An FL-based hybrid deep learning technique is employed in this work. To the best of our knowledge, this research is the first study to successfully identify theft detection using FD, as described above.

1.1. *Contributions and paper organization:*

The graphical abstract of this study has been presented in Fig. 1 and a description of each part has been added in subsequent sections. The main contributions of this study are:

- A novel FL-Convolutional Gated Recurrent Unit (ConvGRU) mixed deep learning model for the identification of theft types is presented in this study. The advantages of GRU networks, convolutional neural networks, and FL are all included in this model. The integration of these techniques provides a powerful framework for accurately classifying theft types.
- The research proposes a federated learning approach for training the FL-ConvGRU model. Federated learning enables collaborative model training without the need to share raw data among different entities. This approach addresses privacy concerns while leveraging the collective intelligence of distributed datasets, making it suitable for scenarios where data privacy is crucial.
- The experiment conducted in this study evaluates the proposed FL-ConvGRU model on diverse structured dataset relevant to theft classification. The dataset was obtained from online repositories and real-time sources, ensuring their variability and real-world applicability. This evaluation demonstrates the model's effectiveness and robustness across different types of theft attempts.
- The paper includes a comprehensive comparative analysis of the proposed FL-ConvGRU model with other popular classifiers, including the base FL-CNN, FL-LSTM, and FL-GRU model. The comparison highlights the superior performance of the FL-ConvGRU model in terms of classification accuracy. This analysis contributes to the validation and credibility of the proposed approach.
- The study provides valuable insights into potential areas for future research in smart meter based theft identification. It suggests exploring further feature selection techniques, optimization methods, addressing time constraints, and employing different data privacy techniques. These future research directions aim to enhance the effectiveness and efficiency of predictive classifiers for diagnosing theft attempts.

2. Evolving power infrastructure: Challenges, innovations, and solutions in the Smart Grid era

In Industry 5.0, the integration of the Smart Grid concept revolutionizes electricity generation, distribution, and consumption. Leveraging advanced technologies like the Internet of Things (IoT), federated learning, big data analytics, and predictive modeling, the Smart Grid creates an efficient and intelligent power infrastructure. This interconnected system enables real-time monitoring, control, and optimization,

Table 1
A detailed literature review of federated learning and non-federated learning based electricity theft detection.

Citation	Year	Technique	Results	Centralized	Privacy preserved
Liao et al. (2023)	2023	Euclidean and Graph CNN	MAP: 0.960 AUC: 0.770	✓	X
Haq et al. (2023)	2023	Deep CNN	Accuracy: 0.940 Recall: 0.970 Precision: 0.930 F1-Score:0.965	✓	X
Yan and Wen (2021a)	2021	Extreme Gradient Boosting	Accuracy: 0.990 Precision: 0.975 Recall: 0.937	✓	X
Li et al. (2019)	2019	Deep Learning and Random Forest	F1-Score: 0.960 Precision: 0.970 Recall: 0.980	✓	X
Lepolesa et al. (2022)	2022	Deep Neural Network	F1-Score: 0.930 Precision: 0.914 Recall: 0.946 Accuracy: 0.916	✓	X
Ashraf et al. (2022)	2022	FL-DNN	F1-Score: 0.887 Precision: 0.890 Recall: 0.916	✓	✓
Wen et al. (2022)	2022	FedDetect	AUC: 0.791 Accuracy: 0.919	✓	✓
Our model	2023	FL-ConvGRU	Accuracy: 0.980 Recall: 0.970 Precision: 0.980 F1-Score: 0.980	✓	✓

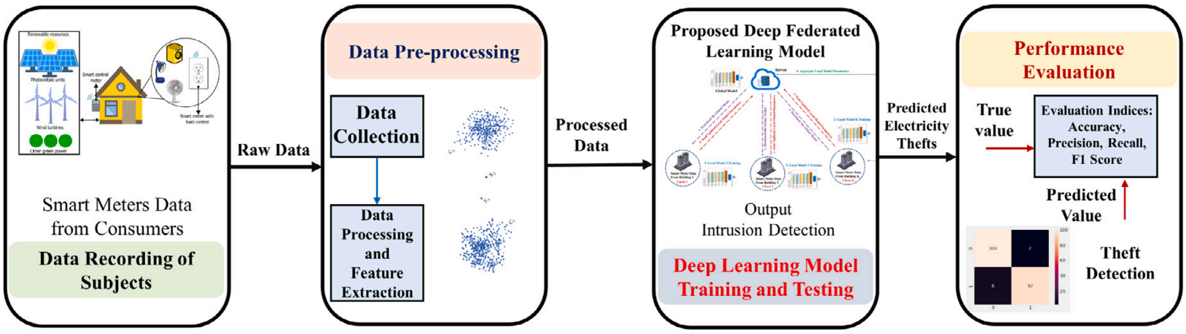
Smart Meters Data
from Consumers
Data Recording of Subjects
Raw Data
Data Pre-processing
Data Collection
Data Processing and Feature Extraction
Processed Data
Proposed Deep Federated Learning Model
Output
Intrusion Detection
Deep Learning Model Training and Testing
Predicted Electricity Thefts
Performance Evaluation
True value
Evaluation Indices: Accuracy, Precision, Recall, F1 Score
Predicted Value
Theft Detection

Fig. 1. An illustration of the suggested Theft Detection method for Smart Grids based on Federated Learning.

ensuring grid stability, load balancing, and fault detection. By integrating distributed energy resources (DERs) and bidirectional energy flows, Smart Grids maximize renewable energy utilization while reducing reliance on fossil fuels. Smart meters, sensors, and demand response programs empower consumers to actively manage energy consumption, promoting efficiency. The Smart Grid in Industry 5.0 represents a sustainable, resilient, and digitally empowered industrial ecosystem.

2.1. Concept of smart grid

The smart grid embodies an upgraded and intelligent electricity infrastructure that utilizes advanced technologies to enhance power systems' efficiency, reliability, and sustainability. It encompasses a wide range of interconnected components, including power generation sources, transmission and distribution networks, consumer devices, and energy management systems (Fang et al., 2011).

At the heart of the smart grid concept is the integration of digital communication, sensing, and control technologies into traditional power grids. This enables real-time monitoring, control, and optimization of energy generation, transmission, and consumption. The key features of the smart grid are shown in Fig. 2. Key features are listed below (Kabalci, 2016):

- Advanced Metering Infrastructure (AMI): enables two-way communication between utilities and consumers through smart meters, facilitating accurate billing, load monitoring, and demand response programs by recording and transmitting energy consumption data at regular intervals.
- Intelligent sensors and Monitoring: devices deployed throughout the smart grid infrastructure collect real-time data on voltage levels, current flows, line conditions, and equipment performance. This data assists utilities in fault detection, grid stability management, and energy flow optimization.
- Automation and Control Systems: play a crucial role in the smart grid, enabling real-time monitoring, analysis, and control of power generation, distribution, and consumption. These systems incorporate automated devices like switches, reclosers, and voltage regulators to optimize grid operations, reduce outages, and swiftly respond to grid disturbances.
- Distributed Energy Resources (DERs): The smart grid integrates DERs like solar panels and wind turbines, enabling efficient utilization of renewable energy and reducing reliance on fossil fuel-based power generation. It achieves this by monitoring and managing the output of these resources.
- Demand Response and Energy Efficiency: The smart grid enables utilities to implement demand response programs, incentivizing consumers to adjust energy consumption during peak periods

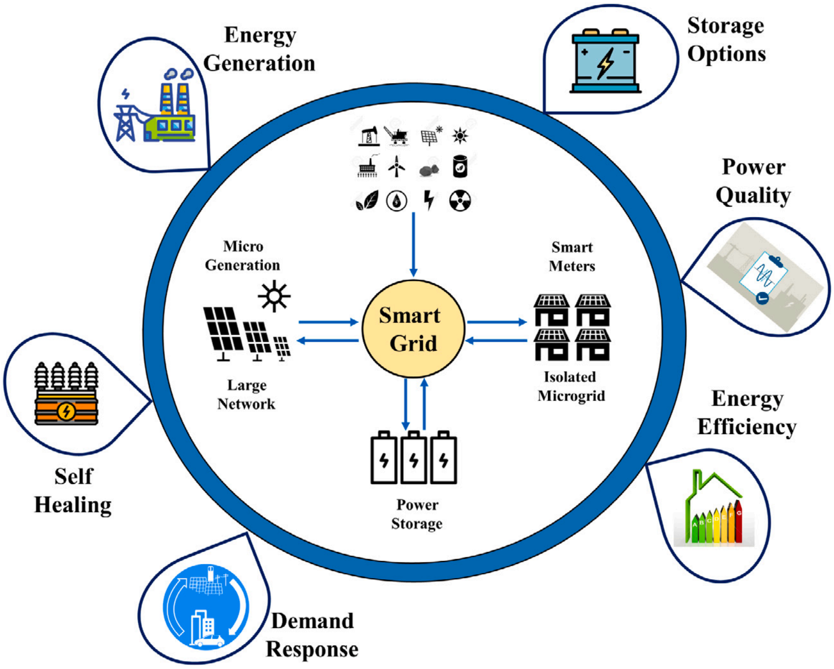

Fig. 2. Overview of the current smart grid's integrated hardware elements and distinguishing characteristics, emphasizing the integration of digital technologies with conventional electricity infrastructure.

to balance supply and demand. Additionally, it promotes energy efficiency by furnishing consumers with real-time energy consumption data, thereby empowering them to make informed choices and curtail their overall energy usage.

- Grid's Resilience and Self-Healing: capabilities minimize disruptions and faults. Through automated monitoring and control systems, the grid swiftly detects and isolates faults, reroutes power flows, and restores service more efficiently. This enhances the reliability of the power grid by reducing outage durations.
- Data Analytics and Predictive Maintenance: The smart grid generates vast data from the power system. Utilities use data analytics and predictive maintenance to analyze patterns, predict failures, and optimize maintenance. This proactive approach reduces downtime, improves asset management, and optimizes performance.

The constant, high-resolution data stream from AMI and smart meters provides a favorable environment for analysis in the context of our study, which focuses on power theft detection. Even if it has many uses, this data might still be vulnerable if not protected. Unauthorized use or power theft might be indicated by anomalies in the data or anomalous trends. We can take advantage of this vast amount of data without sacrificing data privacy by combining FL with our ConvGRU model. FL permits model training across several devices or nodes utilizing their local data, as opposed to centralizing data for analysis, which may reveal flaws. The raw data is not centralized; only the model changes are. This ensures that AMI and smart meter data integrity and confidentiality are not compromised while still enabling in-depth and comprehensive analysis. By using Federated Learning, AMI, and smart meter capabilities, we are laying the groundwork for a smart grid that is both effective and naturally secure, perfectly aligning with the goals of a strong, resilient, and sustainable Industry 5.0 environment.

2.2. Industry 5.0

The Fifth Industrial Revolution, often known as Industry 5.0, is the union of machines and people in an organization (Leng et al., 2022). It essentially describes how people collaborate with intelligent robots and machines, and how cutting-edge technologies like big data and the Internet of Things (IoT) make it possible for people to work more quickly and effectively, giving Industry 4.0's features a more human touch (European Commission, Directorate-General for Research and Innovation, 2021). The European Commission proclaimed the Fifth Industrial Revolution, or Industry 5.0, around 2021 to acknowledge that it will achieve goals in thriving societies in addition to fostering industry growth and employment creation. Industry 5.0 essentially consists of three main strategies:

- Human-centered approach: This value puts the needs and interests of people at the front of the priority list for production.
- Sustainability: Industry 5.0 must take the initiative in preserving the planet's natural resources and creating circular processes that reuse, repurpose, and recycle them while increasing productivity and effectiveness.
- Building resilience entails creating a strong strategy to defend key infrastructure when stranded in a crisis.

In the context of Industry 5.0, the integration of the smart grid plays a crucial role in transforming the way industries operate. Here are the connections between Industry 5.0 and the smart grid:

- Energy Consumption and Optimization: Industry 5.0 integrates industrial processes with the smart grid, enabling real-time insights into energy consumption (Abou Houran et al., 2023). This promotes resource optimization, identifies areas for improvement, and implements demand response strategies to reduce stress on the grid.
- Renewable Energy Integration: Industry 5.0 integrates renewable energy into industrial operations through the smart grid. Real-time monitoring and control of distributed energy resources enable manufacturers to leverage renewable energy generation, reducing reliance on conventional sources.
- Grid Resilience and Stability: The smart grid's bidirectional communication enhances grid resilience and stability. In Industry 5.0, manufacturing facilities act as "prosumers" by generating and supplying excess energy. This dynamic interaction balances supply and demand, improves grid stability, and supports the integration of intermittent renewable energy sources.

- Data-Driven Decision Making: Industry 5.0 utilizes smart grid data to make data-driven decisions, optimizing operations, reducing costs, and improving efficiency.
- Resilient and Adaptive Infrastructure: Industry 5.0 and the smart grid prioritize adaptable and resilient infrastructure. They integrate advanced technologies to monitor, control, and respond to changing conditions, enabling industries to quickly address disruptions and maintain continuous operations.

2.3. *Electricity theft: A looming threat to the smart grid*

Theft of electricity poses a substantial challenge for power utilities and jeopardizes both the robustness and economic sustainability of the smart grid. Depuru et al. (2011). It refers to unauthorized consumption, tampering, or diversion of electricity, leading to revenue losses for utilities and potential safety hazards.

Within the framework of the smart grid, the identification and prevention of electricity theft become more intricate, owing to the intricate nature of the interlinked and digitized power infrastructure. However, smart meters play a crucial role in mitigating electricity theft by providing advanced monitoring and detection capabilities (Xia et al., 2022). Smart meters are useful in addressing this issue according to the following points:

- Consumption Measurement: Smart meters ensure accurate and real-time measurement of electricity consumption. They eliminate the need for manual reading, reducing human errors and manipulation. By recording data at regular intervals, smart meters identify discrepancies between recorded and billed consumption, helping utilities detect potential theft.
- Tamper Detection and Alerts: Smart meters have tamper detection features to identify unauthorized manipulation. They send alerts to utilities when tampering is detected, allowing for swift action against theft.
- Remote Monitoring and Data Analytics: Smart meters allow remote monitoring of energy consumption and provide data for analysis. Utilities utilize advanced analytics to detect anomalies, identify abnormal usage patterns, and flag suspicious activities indicating theft. Continuous monitoring enables the prompt investigation and addressing of potential cases of electricity theft.

Additionally, smart meters reduce non-technical losses by limiting theft, billing errors, and inaccurate readings. Real-time data enables utilities to differentiate legitimate consumption, recover revenue, and invest in grid improvements. Moreover, Smart meters prioritize security with robust features and data encryption. This protects data integrity and prevents unauthorized access. Encryption techniques secure consumption data during transmission, reducing the risk of fraudulent activities and unauthorized manipulation or access to sensitive information.

2.4. *Enhancing security: Reliable methods for theft detection*

Detecting electricity theft is a complex task due to the clandestine nature of the activity and the vast amount of data involved in the smart grid. However, the application of secure and reliable theft detection can be significantly enhanced using a technique called deep federated learning, which leverages smart meter data.

2.4.1. *Obstacles and hurdles in theft detection*

A detailed explanation of the theft detection challenges is elaborated by Yan et al. in Yan and Wen (2021b) and can be categorized into three distinct groups. The first challenge comes from the Data Volume and Variety. Although the smart grids generate massive amounts of data, each system is customized for various local sources generating heterogeneous data. Identifying theft patterns requires advanced techniques capable of handling high volumes and diverse data. Second and the main concern is privacy concerns because of the sensitive information about energy consumption patterns and user behavior. Preserving consumer privacy while detecting theft is crucial to ensure the ethical and legal use of data. Thirdly the distributed Data across numerous locations and utility providers makes Collecting and centralizing all the data for analysis can be logistically challenging and may raise data ownership and sharing concerns. This is where FL shines while providing solutions to all these concerns.

2.4.2. *Deep federated learning*

Deep federated learning is a privacy-preserving deep learning technique that allows the collaborative analysis of decentralized data (Elayan et al., 2021). It enables multiple parties, in this case, utility providers, to jointly train a deep learning model without sharing their raw data. The model is trained locally on each utility's smart meter data while preserving privacy and then aggregated to generate a global model with insights from the collective data.

2.4.3. *Robust methods for ensuring secure theft detection*

Secure and reliable theft detection can be achieved by:

1. Data Localization: Each utility provider retains control over its smart meter data, ensuring data remains localized and reducing privacy risks associated with centralized data storage.
2. Model Collaboration: Utility providers collaborate by sharing only model updates, gradients, or summary statistics instead of raw data. This preserves the privacy of individual consumption patterns while enabling knowledge sharing and model improvements.
3. Privacy-Preserving Techniques: Deep federated learning incorporates privacy-preserving techniques like differential privacy, which adds noise to individual data points, protecting consumer privacy while maintaining the overall accuracy of the model.
4. Robust Model Training: By training on diverse datasets from multiple utilities, the deep federated learning model gains a broader understanding of theft patterns and can detect anomalies more accurately. It can identify common theft indicators, such as abnormal consumption patterns, tampering signatures, or meter bypasses, while accounting for local variations.
5. Real-Time Analysis: Deep federated learning models can continuously update and refine their theft detection capabilities by incorporating new data as it becomes available. This real-time analysis enables prompt identification and response to potential theft incidents.

Deep federated learning from smart meter data provides a secure and reliable approach to theft detection in the smart grid. It addresses challenges related to data volume, privacy concerns, and distributed data sources. By leveraging the collective knowledge embedded in decentralized smart meter data, deep federated learning enables accurate and timely identification of theft patterns while preserving consumer privacy and ensuring the integrity and efficiency of the smart grid.

2.5. *Smart Grid, Industry 5.0 and federated learning*

The integration of multiple power sources, efficiency in energy use, real-time monitoring capabilities, and creating two-way communication between customers and energy suppliers are characteristics of the growth of the Smart Grid, which focuses on multidimensional energy management. Our method, which is based on federated learning (FL), emerges as a crucial tool inside this complex framework. By using FL, we have made sure that intelligence-building on the grid continues to be collaborative while ensuring that data stays decentralized. This fits in nicely with the philosophy of the Smart Grid: decentralized energy sources cooperating to power a single system. We increase the grid's resilience by utilizing FL for theft detection to make sure it is strong

Table 2
General information of theft dataset.

Items	Values
Total no. of data instances	560,655
No. of columns	Numerical columns: 10 Categorical columns: 2 (Including target column) Total columns: 12
Coding technique for categorical column	Label encoder
No. of consumer types (Data column: Class)	16

despite the obstacles of the digital era. This successfully underscores the crucial significance of federated mechanisms in contemporary energy solutions. The digitalization of the Smart Grid is consistent with Industry 5.0's advocacy of the fusion of technology power with human intuition. The grid is now undergoing a transformation, moving from being merely a conduit for electrical current to being an environment-responsive, sentient system. Our suggested theft detection technique, which depends on Federated Learning's capabilities, precisely embodies this Industry 5.0 vision. A successful human-machine symbiosis is ensured by FL, which makes sure that although individual data points stay localized, the collective intelligence of the grid is constantly improved. The federated approach also emphasizes the protection of personal data privacy, a pillar of Industry 5.0, reiterating that even as we develop toward sophisticated technical integrations, the sacredness of human privacy remains inviolable.

3. Dataset description and processing

The data utilized for this project was sourced from the Open Energy Data Initiative (OEDI) portal. It is a centralized repository for high-value energy research datasets gathered from the Programs, Offices, and National Laboratories of the U.S. Department of Energy (Leite and Mantovani, 2016). It comprises of numerous datasets that have been carefully selected in order to speed up accessibility and cooperation. This data lake contains data from a variety of sources, including the corporate sector, academic organizations, and research facilities. Energy use for 16 distinct consumer categories is included in the dataset. The original dataset encompasses energy consumption measurements for diverse consumers spanning a year. Every hour, for a total of 24 h, measurements are taken. About the created dataset, more details are provided in Table 1. The key points of the dataset's description are condensed into summarized elements for clarity and easier reading. The dataset has two categorical columns and ten numerical columns. The dataset included no null values, thus in order to incorporate the category columns and make it usable for training, we used Label Encoder to convert the columns into numerical features that are easier to implement for DL. The Standard-Scalar library, which is accessible in Python, is then used to normalize numerical columns. Utilizing the data values, 4 different scenarios are available:

1. All theft types (target variable) are taken into consideration for classification.
2. The theft 3 type is dropped and the theft classification is changed to six types.
3. The theft 6 type is then removed from the dataset and the classification problem is changed to five theft types.
4. Finally, the Class column in the dataset, which contains data instances from 16 different silos (organizations, research institutes, etc.) and a variety of data instances across each silo.

The procedures for gathering, processing, and potential uses of the data are described. The use of this data seems to hold promise for the development of data-driven algorithms for categorizing different sorts of theft. The general and statistical descriptions of the theft type datasets are depicted in Table 2 and Table 3 respectively. These include no. of data instances, no. of columns, mean, standard deviation and maximum and minimum values.

In multivariate analysis and statistics, a correlation matrix is typically used to explore the relationships between several variables. As can be seen in correlation matrix for this dataset in Fig. 3, all data features have a negative relation with the target column. The distribution of the class instances is shown in Fig. 4.

The two category columns in the dataset are Class and Theft (the target variable). It is essential to be aware of the data examples that are accessible for each type of theft in order to completely comprehend how to build a system for categorizing different types of theft. Additionally, the dataset has a class column that divides the data into various buildings (such as hospitals, schools, and others) and may be used in FL's cross-silo strategy. Therefore, it is crucial to understand the number of data instances present in each building (silo), as well as the availability of theft-related data in each silo.

4. Proposed technique

4.1. Federated learning

Federated learning (FL) is a decentralized method of machine learning (ML) in which training takes place throughout a network of servers or devices rather than in a single location. In this paradigm, the model training happens locally, and the data is kept on local servers or individual devices rather than being collected in a single location (Li et al., 2020). To create a universal model, solely the revised model parameters are aggregated and transmitted across the devices. A cutting-edge method of ML that tackles the issues of data privacy and centralization is federated learning. All of the data is normally gathered in a single server or data repository, where the model is developed, in traditional ML. However, centralizing data frequently leads to worries about security, privacy, and adherence to data protection laws. By allowing ML models to be trained on dispersed devices or local servers, federated learning adopts a decentralized strategy. The data is stored on individual devices, such as smartphones, IoT devices, or edge servers, rather than being sent to a central server. This method has a number of benefits, including more privacy, cheaper communication, and better scalability. Due to the abundance of data available recently, ML and DL-based techniques have experienced enormous growth. The visual overview of the step by step flow and key components of proposed method is shown in Fig. 5. The adopted approach uses the following steps:

- Each participating client receives an initial model from the coordinator first.
- Using their own local datasets, each client independently develops a unique learning model, sending the updated model back to the coordinator for aggregation.
- The aggregated model updates are then provided back to the local participating client after aggregation.
- Until the model converges or the predetermined number of iterations has been reached, this process is repeated. Less communication overhead results from the client–server design.

To minimize the aggregated local loss function $f_a(\theta^a)$, FL seeks to identify the best global model $theta$ in Eq. (2), where x_i is the data-attributes, y_i represents data-labels, n_a points out the local-data size,

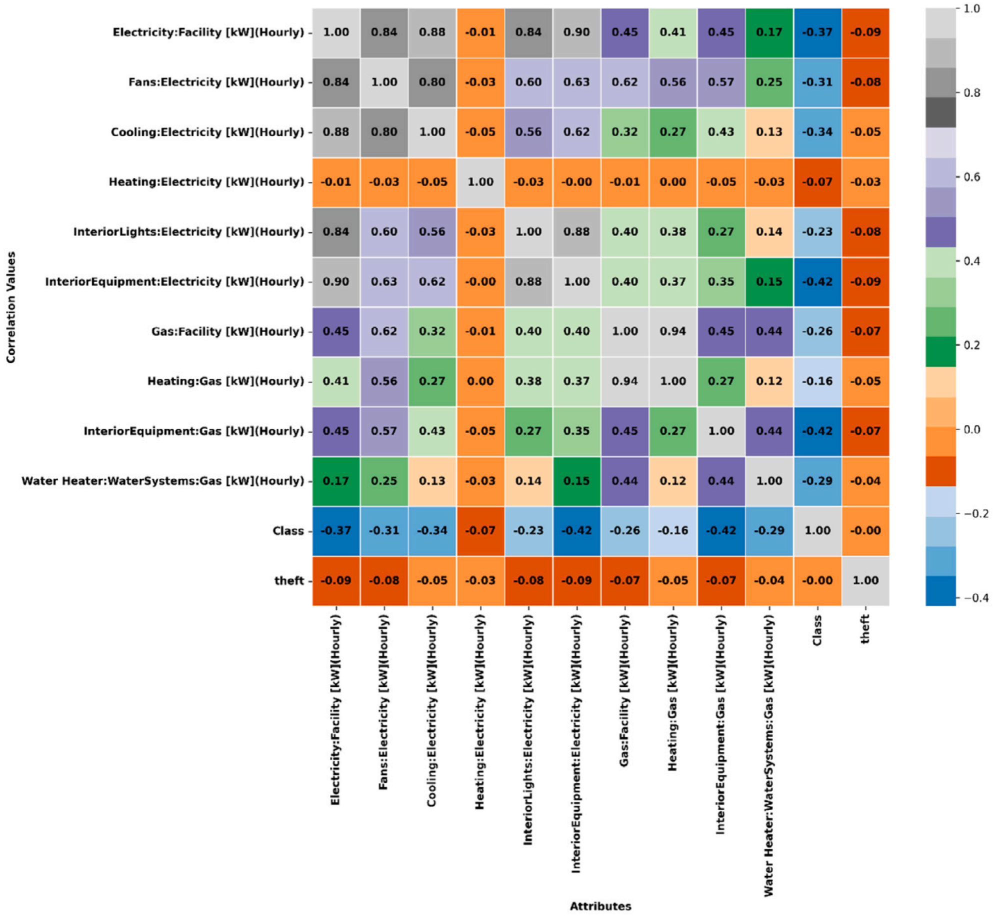

Fig. 3. Correlation matrix of data attributes with inclusion of target variable.

Table 3
Numerical data features information.

Data feature	Mean	Std.	Min	Max
Electricity:Facility [kW](Hourly)	161.77	287.32	0.0	1726.43
Fans:Electricity [kW](Hourly)	13.79	24.08	0.0	240.01
Cooling:Electricity [kW](Hourly)	43.77	117.10	0.0	890.62
Heating:Electricity [kW](Hourly)	0.84	6.12	0.0	277.99
InteriorLights:Electricity [kW](Hourly)	32.98	65.17	0.0	448.56
InteriorEquipment:Electricity [kW](Hourly)	42.63	73.49	0.0	448.56
Gas:Facility [kW](Hourly)	77.31	178.68	0.0	4491.65
Heating:Gas [kW](Hourly)	53.91	157.72	0.0	4480.73
InteriorEquipment:Gas [kW](Hourly)	8.16	15.95	0.0	91.79
Water Heater:WaterSystems:Gas [kW](Hourly)	15.23	52.63	0.0	783.87

and $n = \sum_{k=1}^{C\times A} n_a$ all local clients do not contribute in consecutive iterations.

$$f_a(\theta^a) = \frac{1}{n_a}\sum_{i=1}^{n_a} l(x_i, y_i; \theta^a) \tag{1}$$

where k represents the index of the client among n_a sample pairings, and l denotes the loss function.

$$\min_{\theta} \sum_{k=1}^{C\times A} \frac{n_a}{n} f_a(\theta^a) \tag{2}$$

According to the features of data distribution among the linked customers, as initially specified in the study (Zhang et al., 2021), FL may be broadly divided into three categories: horizontal, vertical, and transfer FL. In this paper, HFL is used largely, also including the concept from both Cross-Silo and Cross device approaches of FL.

4.1.1. Horizontal FL (HFL)

Homogeneous FL (Yang et al., 2019), Also referred to as horizontal federated learning (FL), this approach pertains to situations where participating customers' training data share the same feature space but possess distinct sample spaces. To illustrate, consider a simple example: Clients 1 and 2 each have unique rows of data with identical personal attributes, where each row represents details for an individual. The FedAvg method (McMahan et al., 2017) is a representative HFL algorithm (Algori 1) that represents pseudo code for this process, As illustrated by Zhu et al. (2020) performance varies with low participation rate C where $m = C \times A$ is clients number. The same model architecture featuring distinct parameter values can be observed within both the global model θ_t (where t represents the communication round) and all local models θ_a. In is scenario aggregated model improves with each local model $theta_a$ training performance according to the ratio n_a/n proportional to the client data in global model $theta_t$. This is done using a batch size of B and a learning rate of η.

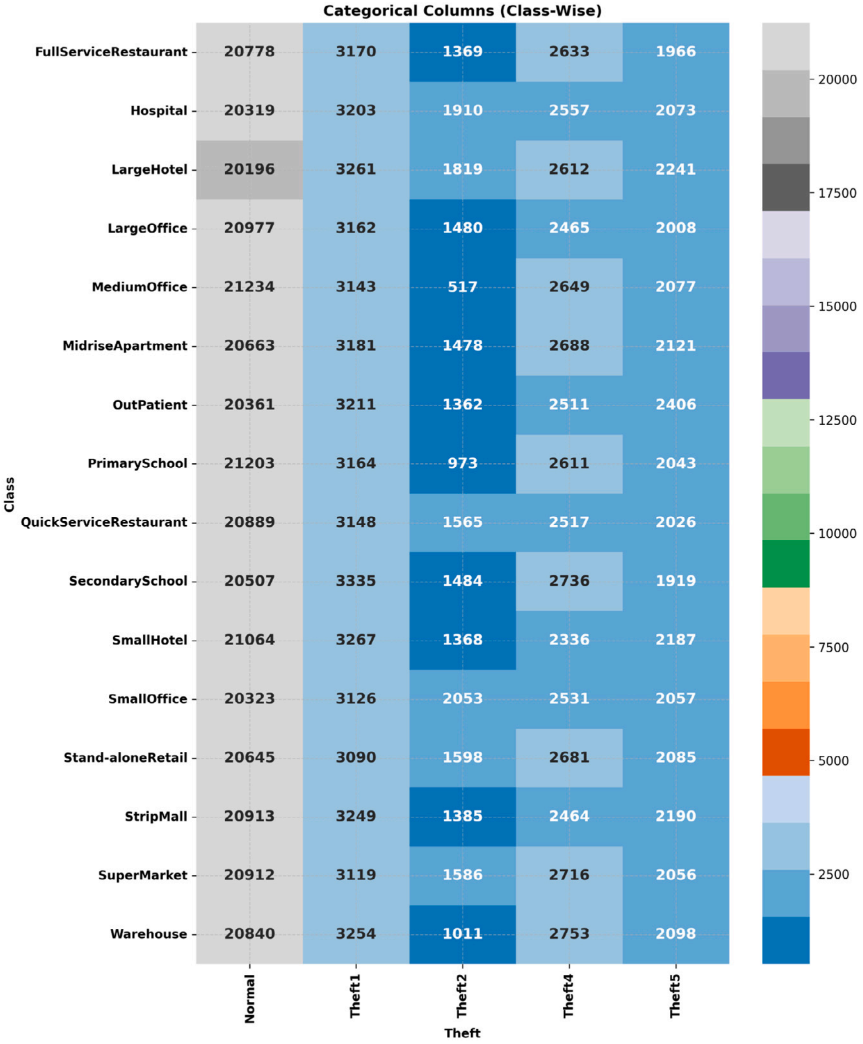

Fig. 4. Distribution of Class Instances in the dataset.

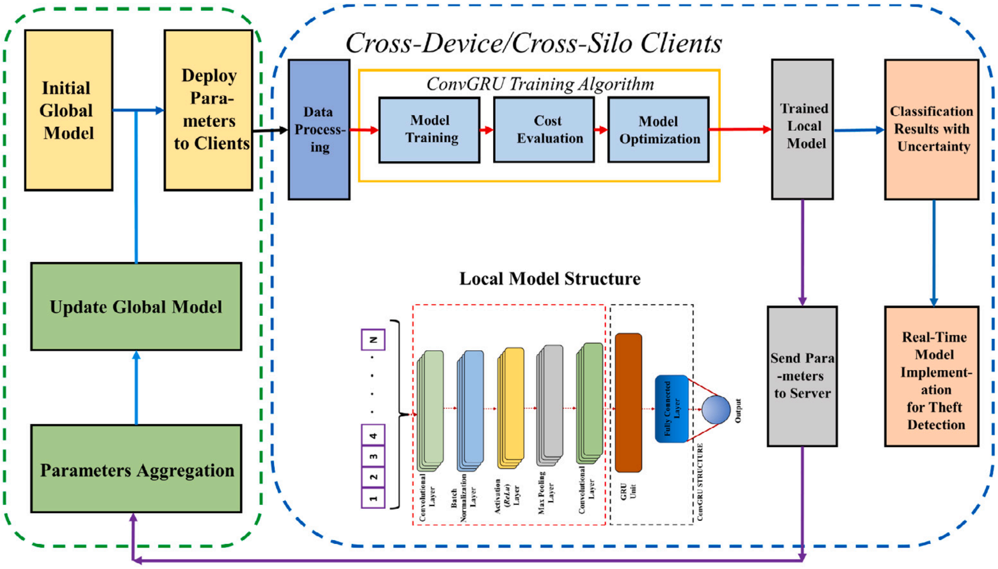

Fig. 5. Visual overview of the step-by-step flow and key components of the proposed model.

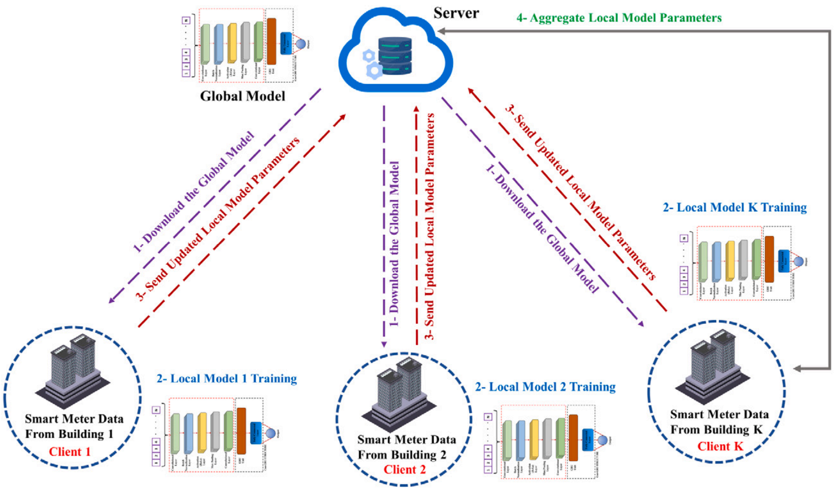
Server
4- Aggregate Local Model Parameters
Global Model
1- Download the Global Model
3- Send Updated Local Model Parameters
1- Download the Global Model
3- Send Updated Local Model Parameters
3- Send Updated Local Model Parameters
1- Download the Global Model
2- Local Model K Training
2- Local Model 1 Training
Smart Meter Data
From Building 1
Client 1
Smart Meter Data
From Building 2
Client 2
2- Local Model 2 Training
Smart Meter Data
From Building K
Client K

Fig. 6. Architecture of the Cross-Silo Federated Learning (FL) setup.

Algorithm 1 : FedAvg

```
Initialize $A$ to denote the total number of clients, $B$ for the mini-batch size, $T$ to represent the total number of communication rounds, $E$ as the total number of training epochs, and $\eta$ as the learning rate.
Server:
Initialize $\theta_0$ as the global model
while $t \leq T$ do
    Select $m = C \times A$ clients, given that $C \in (0, 1)$
    for Client: $a = 1, 2, \ldots, m$ in parallel do
        Transfer $\theta_t$ to Client $a$
        Update Client $a$ using Algorithm 2 and receive $\theta_a$.
        Update global model $\theta = \frac{1}{n} \sum_{k=1}^{C\times A} n_a \theta_a$
        t = t + 1
end while
return  Global Model
```

Algorithm 2 : Client a Update

```
Replace local model $\theta_a$ with $\theta_t$
for Local epoch $\in 1 : E$ do
    for Batch $b \in 1 : B$ do
        $\theta_a \leftarrow \theta_a - \eta \nabla L_a(\theta_a, b)$
    end for
    return  $\theta_a$
end for
```

The FL system's performance may typically be improved to some extent by adjusting E, B, and eta appropriately for the job. This can be found through sensitivity analysis or optimized using metaheuristic search techniques (Zeng et al., 2021).

Horizontal FL offers a straightforward and effective solution to prevent private local data leakage. Communication between the server and clients is limited to the global model parameters (θ_t) and local model parameters (θ_a), ensuring that the training data remains solely on the client devices and inaccessible to other parties.

4.1.2. Cross-Silo FL

When a smaller number of participating devices are available and remain accessible throughout all rounds, Cross-Silo Federated Learning is utilized. The training data may be in FL format, either horizontally or vertically. Cross-silo is mostly utilized for instances involving organizations. Cross-silo FL is used by works like (Zhang et al., 2020) to create their model. In this paper, using the categorical column (Class) in the theft dataset, which comprises various organization names, a cross-silo technique for theft detection is created. We construct multiple data silos, each representing a different data source, by considering each organization as a separate client. Similar to HFL, where training data from participating consumers have the same feature space but a distinct sample space, each organization has its own subset of theft data with the same personal qualities. Local models are separately taught inside each organization utilizing the horizontal federated learning strategy. These regional models accurately depict the theft patterns and traits that are unique to each organization. The local models from each organization are then combined to build the global theft detection model, using the information provided across the silos. This strategy protects the security and privacy of the data that each organization has while enabling joint training of an extensive theft detection model. The proposed cross-silo structure of FL is shown in Fig. 6.

4.1.3. Cross-device FL

Cross-device Federated Learning is implemented in scenarios involving diverse devices. To facilitate this form of FL, it becomes crucial to adopt sustainable approaches such as client selection and incentive schemes (Yu et al., 2020). Additionally in this study, we have created a cross-device strategy utilizing theft dataset with cross-silo approach. We allow the training of local models on each device by randomly partitioning the dataset into 10 clients representing various devices, such as smartphones, IoT sensors, and security cameras. The devices in question employ horizontal federated learning and share a feature space, but they each retain different samples of theft data. Using its unique dataset, each device separately trains a local model. The device-specific theft patterns and features are captured by these local models. The local models from each device are then combined to build the global theft detection model, which incorporates the knowledge gleaned from the variety of devices. This method ensures the privacy and security of the data stored on individual devices while enabling a thorough knowledge of theft detection that takes into account the specifics of each device's data.

In cross-silo and cross-device scenarios, we use horizontal federated learning to handle privacy and data security challenges related to theft detection. Each organization functions as a separate client in the cross-silo environment, supplying its distinct theft-related data to identify silo-specific trends. Each device functions as a client in the cross-device configuration to collect device-specific patterns. A strong theft detection model that takes use of the collective knowledge across organizations and devices is produced by the federated learning process, which enables the aggregation of local models from both cross-silo and cross-device settings. By using an integrated strategy, theft detection is more accurate and efficient while protecting the privacy and security of sensitive information stored by each organization and device.

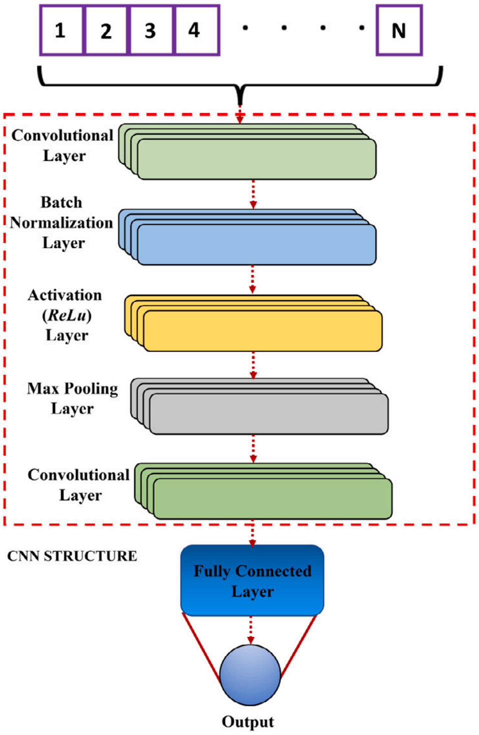

Fig. 7. Architecture of the Convolutional Neural Network (CNN).

4.2. CNN model

Convolutional Neural Networks (CNNs) have demonstrated remarkable efficacy in a range of computer vision assignments, including image classification and object detection. Nevertheless, adapting them for sequential data like time series or 1D signals necessitates certain adjustments. The 1D CNN model is a variant of CNNs specifically designed to process 1D input data (Ozcanli and Baysal, 2022).

The 1DCNN model consists of several key components: input layer, convolutional layers, pooling layers, fully connected layers, and output layer.

4.2.1. Input layer

The input layer of the 1DCNN model receives the 1D input signal, which can be represented as a sequence of values $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, where x_i is the value at position i. The input signal is typically represented as a 1D array or a time series.

4.2.2. Convolutional layers

The convolutional layers perform feature extraction by applying a set of filters or kernels to the input signal as shown in Fig. 7. Each filter is a small window of size k, which slides across the input signal with a specified stride, performing element-wise multiplications and additions. The output of the convolution operation is computed as:

$$\mathbf{c}_i = f(\mathbf{W} \cdot \mathbf{x}_{i:i+k-1} + b), \tag{3}$$

where $\mathbf{W}$ is the weight matrix, $\mathbf{x}_{i:i+k-1}$ is the input subsequence from position i to $i+k-1$, b is the bias term, and $f(\cdot)$ is the activation function, such as ReLU (Rectified Linear Unit).

4.2.3. Pooling layers

Pooling layers are employed to diminish the dimensionality of feature maps generated by convolutional layers. Among these, max pooling is the most common, involving the selection of the maximum value within a predetermined window. Max pooling helps in capturing the most salient features and provides translation invariance. The pooling operation can be defined as:

$$\mathbf{p}_i = \max(\mathbf{c}_{i:i+s-1}), \tag{4}$$

where $\mathbf{p}_i$ is the pooled value at position i and s is the pooling window size.

4.2.4. Fully connected layers

The fully connected layers collect the learnt features and carry out the final classification or regression after extraction of features and reduction of dimensionality. Each neuron in the layer with complete connectivity is linked to every neuron in the layer below. The output of the fully connected layer can be calculated as:

$$\mathbf{h} = f(\mathbf{W} \cdot \mathbf{p} + b), \tag{5}$$

where $\mathbf{W}$ is the weight matrix, $\mathbf{p}$ is the input vector from the previous layer, b is the bias term, and $f(\cdot)$ is the activation function.

4.2.5. Output layer

The ultimate predictions or estimates are generated by the output layer of the 1DCNN model, relying on the acquired features from preceding layers. The quantity of neurons within the output layer is contingent upon the number output parameters for the problem at hand.

4.3. GRU model

RNNs find extensive application in tasks involving sequential data processing, like natural language processing and time series analysis. Nonetheless, conventional RNNs encounter the vanishing gradient challenge, limiting their capacity to capture long-range dependencies. The GRU model, an extension of RNNs, tackles this concern through the utilization of gating mechanisms (Subramanian et al., 2022).

Comprising a series of interconnected recurrent units, the GRU model adjusts its hidden state by considering input data and the prior hidden state. This model incorporates two gating mechanisms as shown in Fig. 8, namely the update gate and the reset gate, which manage information flow and alleviate the challenge posed by the vanishing gradient issue.

4.3.1. Update gate

Designated as z_t, the update gate assesses the fraction of the previous hidden state to retain and merge with the present input. Its calculation employs the sigmoid activation function:

$$z_t = \sigma(\mathbf{W}_z \cdot \mathbf{x}_t + \mathbf{U}_z \cdot \mathbf{h}_{t-1} + \mathbf{b}_z), \tag{6}$$

where $\mathbf{x}_t$ is the input at time step t, $\mathbf{h}_{t-1}$ is the previous hidden state, $\mathbf{W}_z$ and $\mathbf{U}_z$ are weight matrices, $\mathbf{b}_z$ is the bias term, and $\sigma(\cdot)$ is the sigmoid function.

4.3.2. Reset gate

When computing the new hidden state, the reset gate, indicated as r_t, decides how much of the old hidden state should be ignored. It is also computed using the sigmoid activation function:

$$r_t = \sigma(\mathbf{W}_r \cdot \mathbf{x}_t + \mathbf{U}_r \cdot \mathbf{h}_{t-1} + \mathbf{b}_r). \tag{7}$$

where $\mathbf{x}_t$ is the input at time step t, $\mathbf{h}_{t-1}$ is the previous hidden state, $\mathbf{W}_r$ and $\mathbf{U}_r$ are weight matrices, $\mathbf{b}_r$ is the bias term, and $\sigma(\cdot)$ is the sigmoid function.

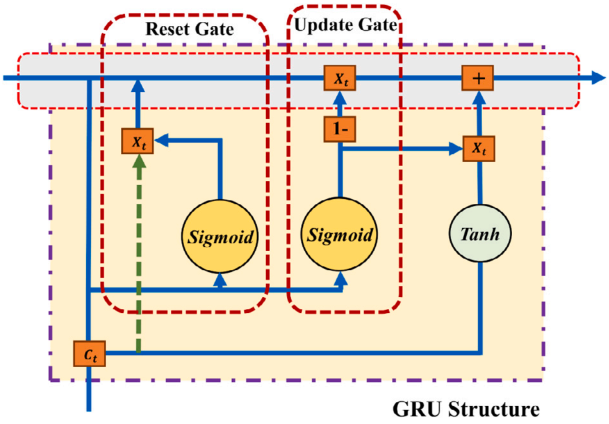

Fig. 8. Architecture of the Gated Recurrent Unit (GRU).

4.3.3. *Candidate hidden state*

The candidate hidden state, denoted as $\tilde{\mathbf{h}}_t$, is a proposed update to the hidden state. It is computed using the hyperbolic tangent activation function:

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \cdot \mathbf{x}_t + \mathbf{U}_h \cdot (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h), \tag{8}$$

where $\mathbf{x}_t$ is the input at time step t, $\mathbf{h}_{t-1}$ is the previous hidden state, $\mathbf{r}_t$ is the reset gate, $\mathbf{W}_h$ and $\mathbf{U}_h$ are weight matrices.

4.4. *ConvGRU*

The ConvGRU model is a hybrid architecture that combines the strengths of CNN in capturing local features and GRU in modeling temporal dependencies. The architecture of the ConvGRU is shown in Fig. 9. This model is particularly effective for sequential data processing tasks, such as speech recognition and video analysis. Comprising of two primary elements, the ConvGRU model integrates convolutional layers for feature extraction from input data and GRU layers for capturing temporal dependencies.

4.4.1. *Convolutional layers*

The convolutional layers in the ConvGRU model extract local features from the input sequence. Each convolutional layer applies a set of filters to the input, which slide across the input sequence and perform convolutions. The output feature maps capture different aspects of the input sequence.

Let $\mathbf{x}_t$ denote the input at time step t, and $\mathbf{W}$, $\mathbf{b}$ be the weight matrix and bias term of a convolutional filter, respectively. The output feature map $\mathbf{c}_{t,i}$ at time step t and filter index i is computed as follows:

$$\mathbf{c}_{t,i} = f(\mathbf{W}_i * \mathbf{x}_t + \mathbf{b}_i), \tag{9}$$

where $*$ denotes the convolution operation, $f(\cdot)$ is the activation function, and $\mathbf{W}_i$ and $\mathbf{b}_i$ are the weight matrix and bias term specific to filter i.

4.4.2. *GRU layers*

The GRU layers in the ConvGRU model capture the temporal dependencies in the input sequence using gated units. Each GRU unit updates its hidden state based on the previous hidden state and the current input.

Let $\mathbf{h}_{t-1}$ denote the hidden state at time step $t-1$, and $\mathbf{W}_z, \mathbf{U}_z, \mathbf{b}_z, \mathbf{W}_r, \mathbf{U}_r, \mathbf{b}_r, \mathbf{W}_h, \mathbf{U}_h, \mathbf{b}_h$ be the weight matrices and bias terms of the GRU unit.

The update gate z_t, reset gate r_t, and candidate hidden state $\tilde{\mathbf{h}}_t$ are calculated as follows:

$$z_t = \sigma(\mathbf{W}_z \cdot \mathbf{c}_t + \mathbf{U}_z \cdot \mathbf{h}_{t-1} + \mathbf{b}_z) \tag{10}$$

$$r_t = \sigma(\mathbf{W}_r \cdot \mathbf{c}_t + \mathbf{U}_r \cdot \mathbf{h}_{t-1} + \mathbf{b}_r) \tag{11}$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \cdot \mathbf{c}_t + \mathbf{U}_h \cdot (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \tag{12}$$

In these equations, $\mathbf{c}_t$ represents the input feature map at time step t, $\mathbf{U}_h$ is the weight matrix associated with the reset gate, $\mathbf{r}_t$ denotes the reset gate, and $\odot$ denotes element-wise multiplication.

4.5. *Hyperparameters of ConvGRU model*

The hyperparameters of the ConvGRU model are essential settings that determine its architecture and behavior. In this section, we elaborate on the hyperparameters and provide typical ranges for each parameter.

4.5.1. *Number of convolutional layers*

The depth of the feature extraction process is dictated by the quantity of convolutional layers.

Range: Typically, 1 to 5 convolutional layers are used, depending on the complexity of the task and the dataset.

4.5.2. *Number of filters*

The depth of the feature extraction process is dictated by the quantity of convolutional layers.

Range: The number of filters can vary widely, usually ranging from 16 to 512 or even higher, depending on the complexity of the task and the input data.

4.5.3. *Convolutional kernel size*

The size of the convolutional kernels specifies the receptive field or the local context captured by each filter.

Range: Common kernel sizes include 3, 5, or 7, representing a window of 3 × 1, 5 × 1, or 7 × 1 over the input sequence.

4.5.4. *Pooling*

Pooling layers decrease the spatial dimensions of feature maps while capturing the most significant information.

Range: Max pooling is commonly used with pooling window sizes ranging from 2 to 4.

4.5.5. *Number of GRU layers*

The number of GRU layers determines the depth of the temporal modeling process.

Range: Usually, 1 to 3 GRU layers are used, depending on the complexity of the task and the dataset.

4.5.6. *Hidden state dimension*

The hidden state dimension defines the number of memory cells or units in the GRU layers.

Range: The hidden state dimension can vary widely, typically ranging from 32 to 1024 or higher, depending on the complexity of the task and the dataset.

4.5.7. *Learning rate*

The learning rate controls the step size during the optimization process, influencing how quickly the model learns and ranges from 0.001 to 0.1. The optimal learning rate depends on the specific task and dataset and is chosen in the ablation study of the architecture an is 0.001 in our study.

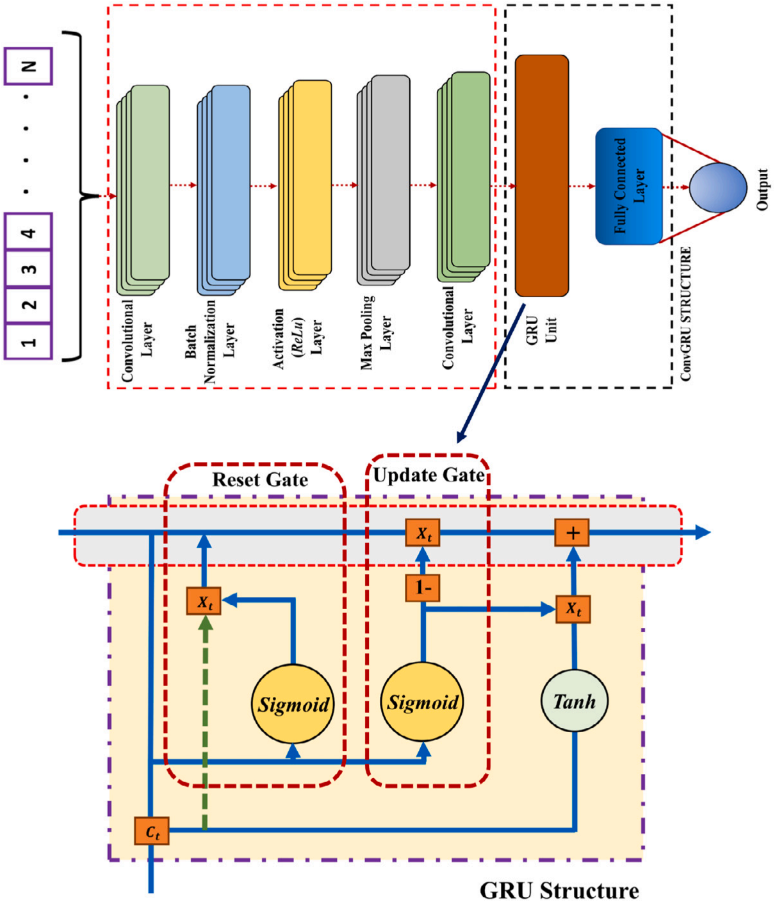

Fig. 9. Architecture of the ConvGRU (Convolutional Gated Recurrent Unit).

Table 4
Hyperparameters and initialization ranges.

Optimized component	Hyperparameters	Range (To initialize)
Convolutional layers	No. of units in 3 layers Filter size in each layer Activation	[2^0–2^9] [1–7] [‘LeakyReLU’, ‘ReLU’, ‘Tanh’]
GRU layers	No. of hidden nodes/neurons	[10–500]
Dense layer	Nodes	[10–500]
Learning configuration	Learning rate Dropout rate	[10^{-5}–10^{-1}] [0, 0.7]

4.5.8. Dropout rate

Dropout serves as a regularization technique that combats overfitting by randomly deactivating a portion of units during training. Dropout rates typically vary between 0.1 and 0.5, reflecting the proportion of units excluded during training. A higher dropout rate (0.5) in our case helps to adapt the system to long-term and short-term feature learning. It is important to note that the optimal values of these hyperparameters can vary depending on the specific task, dataset, and available computational resources. Hyperparameter tuning techniques, such as grid search or random search, can be employed to find the best hyperparameter values for a particular task. Table 4 provides the details of hyperparameters utilized in this study.

4.6. Deep FL based ConvGRU model

A ConvGRU model for deep horizontal federated learning is used in our research study. CNN and GRU, two common deep learning architectures, are combined in the ConvGRU model. The model's 1DConv layer is in charge of removing regional patterns and characteristics from the input data. It employs a collection of trainable filters to the input data to capture spatial and local interdependence. Local patterns in sequential data, such as time series or text data, can be found with remarkable success using this layer. Recurrent connections introduced by the GRU layer, on the other hand, allow the model to recognize temporal relationships and enduring patterns in the input. GRU units feature gating mechanisms that regulate the information flow, enabling the model to gradually forget certain information while updating other information. The GRU layer is hence ideal for problems involving long-term dependent sequential data. The ConvGRU model can detect local and temporal trends in the theft dataset by merging the 1DConv and GRU layers. Local characteristics are extracted from the data by the 1DConv layer, and temporal dependencies and sequential correlations are recorded by the GRU layer. Each participating client in the deep horizontal federated learning process trains its own ConvGRU model using a local dataset. The local models acquire the ability to identify pertinent characteristics and patterns unique to each client's data. Following local training, a new global ConvGRU model is produced by combining the updated model parameters from each client. This

Table 5
Optimized hyperparameters.

Optimized component	Hyperparameters	Optimized values
Convolutional layers	No. of units in 1st layer	200
	No. of units in 2nd layer	1
	No. of units in 3rd layer	3
	Activation	'ReLU'
GRU layers	No. of hidden nodes/neurons in 1st layer	100
	No. of hidden nodes/neurons in 2nd layer	50
	No. of hidden nodes/neurons in 3rd layer	25
Dense layer	Nodes	50
Learning Configuration	Learning Rate	10^{-2}
	Dropout Rate	0.5

worldwide model is an amalgamation of theft trends that have been identified by all involved clients. In deep horizontal federated learning, the ConvGRU paradigm has several benefits. Sequential data, which is frequently used in theft detection activities, may be handled well. The model can learn complicated stealing patterns and linkages since it can collect local and temporal trends. Additionally, because federated learning is dispersed and the training data is stored locally on servers or devices, privacy and security are guaranteed. A potent architecture for deep horizontal federated learning in theft detection is the ConvGRU model. Using a federated learning method, it combines the advantages of 1DConv for local feature extraction and GRU for capturing temporal correlations, allowing the model to learn complex stealing patterns while protecting data privacy.

4.6.1. Halving randomized search

To select the hyperparameter several techniques have been introduced in the literature. The Grid Search method (Ogunsanya et al., 2023) exhaustively searches all possible combinations of hyperparameter values. This process can be time-intensive and computationally demanding, particularly when dealing with an extensive array of hyperparameters and potential values. Alternatively, the Random Search method (Villalobos-Arias and Quesada-López, 2021) randomly selects combinations of hyperparameter values to evaluate. It is less computationally expensive than grid search and can often find good hyperparameter values with fewer evaluations. Most recently the Bayesian Optimization (Eggensperger et al., 2013). This method uses a probabilistic approach for best hyperparameter configurations. It can be more efficient than random search and grid search, especially with a limited budget of evaluations (Bergstra and Bengio, 2012). To tune the hyperparameters of a ConvGRU, we adopt these steps which are also shown in Fig. 10, and define the hyperparameters to tune as follows:

- Learning rate: The rate at which the model adjusts its weights during training.
- Batch size: The number of samples processed in each training iteration.
- Number of layers: The depth of the convolutional GRU network.
- Number of filters: The number of filters used in each convolutional layer.
- Dropout rate: The proportion of inputs randomly set to 0 during training to prevent overfitting.
- Activation function: The non-linear function is applied to the output of each neuron.
- Implement the hyperparameter tuning process:
- Split the dataset into training and validation sets.
- Define the search space for each hyperparameter. For example, specify a range of values for the learning rate or a set of possible values for the number of layers.
- Use a hyperparameter tuning library or framework to perform the search.
- Define a function that builds and trains the Convolutional GRU model with the specified hyperparameters.
- Set up the hyperparameter search algorithm, specifying the search space, the number of iterations, and the evaluation metric to optimize (e.g., accuracy or loss).
- Run the hyperparameter search process, which will evaluate different combinations of hyperparameters and select the best ones based on the specified evaluation metric.
- Retrieve the best hyperparameters found during the search and use them to train a final Convolutional GRU model on the full training set.

Tuning a ConvGRU involves adjusting various hyperparameters related to both the GRU and the convolutional layers. Key hyperparameters include the number of filters, kernel size, stride, padding, and activation function for the convolutional layers, and the number of hidden units, activation function, dropout rate, and recurrent dropout rate for the GRU layers. Learning hyperparameters such as the learning rate, batch size, and number of epochs also need to be tuned. To tune these hyperparameters, you can use techniques like grid search, random search, or more advanced methods like Bayesian optimization or evolutionary algorithms. The goal is to find the optimal combination of hyperparameters that results in improved performance on a validation set or using cross-validation techniques. When tuning the hyperparameters of a ConvGRU, we start by defining the hyperparameters to tune. These include the learning rate, batch size, number of layers, number of filters, dropout rate, and activation function. Next, we incorporate a hyperparameter tuning method. Classical methods such as grid search perform exhaustively searches for all possible combinations of hyperparameter values. The random search methods eliminate the brute force by randomly selecting combinations of hyperparameter values to evaluate. More advanced methods such as Bayesian optimization, use a probabilistic model to estimate the performance of different hyperparameter configurations and select the most accurate to evaluate. After defining the hyperparameters and choosing a tuning method, we implement the hyperparameter tuning process. This involves splitting dataset into training and validation sets, defining the search space for each hyperparameter, using a hyperparameter tuning library or framework to perform the search, defining a function that builds and trains the ConvGRU model with the specified hyperparameters, setting up the hyperparameter search algorithm, running the hyperparameter search process, retrieving the best hyperparameters found during the search, and using them to train a final ConvGRU model on the full training set. Finally, we evaluate the final model on a separate test set to assess its performance (Nagaraj and Malagi, 2023). Halving randomized search leverages both traditional and probabilistic techniques to achieve time and computational efficiency, facilitating swift hyperparameter optimization. In successive halving, the reduction in candidate models and the expansion of training cases between iterations are governed by exponential functions. These exponential functions are influenced by the number of successive halving iterations (Soper, 2022). To elaborate, in each iteration of successive halving, a subset of the most promising candidate models is selected based on their performance. The number of candidate models to be carried forward to the next iteration is typically a fraction of the total number of models evaluated in the

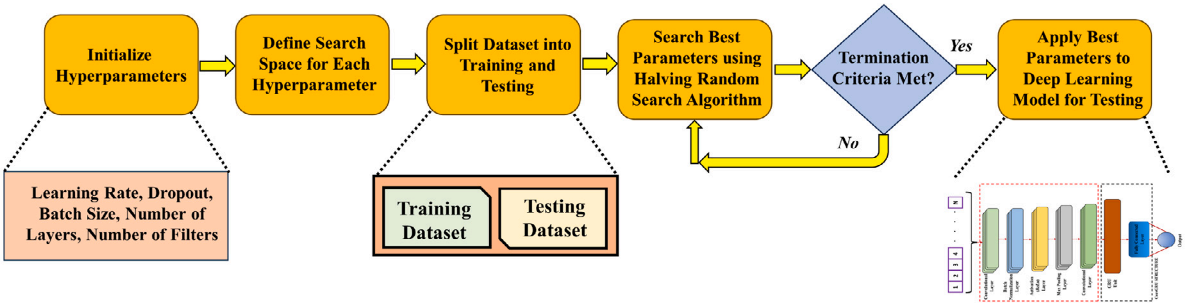
Initialize Hyperparameters
Define Search Space for Each Hyperparameter
Split Dataset into Training and Testing
Search Best Parameters using Halving Random Search Algorithm
Termination Criteria Met?
Yes
No
Apply Best Parameters to Deep Learning Model for Testing
Learning Rate, Dropout, Batch Size, Number of Layers, Number of Filters
Training Dataset
Testing Dataset

Fig. 10. Proposed halving randomized search algorithm-based hyperparameter tuning of ConvGRU Model.

current iteration. This fraction is often referred to as the "reduction factor". Calculating the exponential factor entails establishing the requisite number of iterations for the successive halving procedure. For each iteration of successive halving, the minimum number of iterations (N_{iter}), the maximum number of cases, and the halving factor (h) are determined by selecting approximately 1/h relative to the preceding iterative run, as expressed in Eq. (13).

In order to compute the parameters of the exponential function, it is essential to ascertain the necessary iterations for the halving protocol. With each iteration, a fraction of $1/h$ of the models is retained from the preceding one. The overall number of iterations (N_{iter}) is contingent upon the minimum and maximum instances per iteration, as well as the halving factor (h), and this relationship is depicted in Eq. (13):

$$N_{\text{iter}} = \left\lfloor \log_h \left(\frac{N_{\text{max}}}{N_{\text{min}}} \right) \right\rfloor + 1 \tag{13}$$

As an important factor among hyperparameters after iteration number is the number of Learning models and is calculated by Eq. (14):

$$y = a \cdot e^{x \cdot b}, \tag{14}$$

where each iteration i ($i = 0$) (N_{models}) ML models required for the next iteration are obtained by Eq. (15) (N_{cases}) number of training cases for ongoing iteration as given by Eq. (17):

$$N_{\text{models}} = n(M) \cdot e^{-(i+1) \cdot b_{\text{models}}} \tag{15}$$

$$b_{\text{models}} = \frac{\ln\left(\frac{2}{n(M)}\right)}{-N_{\text{iter}} + 1} \tag{16}$$

$$N_{\text{cases}} = N_{\text{min}} \cdot e^{i \cdot b_{\text{cases}}} \tag{17}$$

$$b_{\text{cases}} = \frac{\ln\left(\frac{N_{\text{max}}}{N_{\text{min}}}\right)}{N_{\text{iter}} - 1} \tag{18}$$

where $n(M)$ is the cardinality of M (selected candidate models). Once the training cases in (N_{cases}) are drawn from the data using Eq. (17). The random sample of N_{cases} are subdivided into k folds. Eq. (15) determines the count of candidate machine learning models (N_{models}) which serve as the input for the subsequent iteration. I the final and current iteration are the same, then N_{models} is equal to 1, which indicates only one best-performing model N_{models} is returned. The standard successive halving algorithm is distinguished on this point With the standard successive halving (SHA) cross-validation is performed to counter-verify. For identified hyperparameters, SHA generates N_{models} best-performing model that propagates in the next iteration. The optimized parameters have been listed in Table 5.

5. Results and discussion

5.1. Evaluation metrics

When addressing electricity theft detection through deep learning models, it becomes imperative to gauge the classification outcomes' performance. Various evaluation metrics can be employed to appraise the precision and efficacy of these models. The commonly used metrics include Accuracy, Precision, Recall, and F1 score. The summary of these results is given in Fig. 11 and Fig. 12.

5.1.1. Accuracy (ACC)

By measuring the proportion of cases that are properly categorized to all instances, accuracy assesses the classification model's overall correctness. It provides a general overview of the model's performance:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{19}$$

where TP (True Positive) denotes instances correctly classified as electricity theft, TN (True Negative) signifies instances correctly classified as non-electricity theft, FP (False Positive) refers to instances incorrectly classified as electricity theft, and FN (False Negative) indicates instances incorrectly classified as non-electricity theft.

5.1.2. Precision (P)

Precision measures the proportion of instances correctly classified as electricity theft out of the total instances classified as electricity theft. It focuses on the accuracy of positive predictions:

$$P = \frac{TP}{TP + FP}, \tag{20}$$

5.1.3. Recall (R)

Recall, often referred to as sensitivity or true positive rate, quantifies the percentage of cases that are accurately identified as power theft out of all instances where electricity theft really occurred. It focuses on capturing the positive instances effectively:

$$R = \frac{TP}{TP + FN}, \tag{21}$$

5.1.4. F1 score

The F1 score represents the harmonic average of precision and recall, offering a well-rounded assessment of the model's effectiveness by taking into account both precision and recall:

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{22}$$

5.2. Detection comparison for known users

As shown in Table 6, it is evident that the DFL-ConvGRU model outperforms the other theft detection technologies, namely DFL-GRU, DFL-LSTM, and DFL-CNN. The DFL-ConvGRU model achieves an accuracy of 0.9778, surpassing the other models' accuracies, which range from 0.7194 to 0.8951. This indicates that DFL-ConvGRU can more accurately classify theft incidents, making it a highly reliable choice for theft detection applications.

Furthermore, the precision of DFL-ConvGRU is reported as 0.9804, significantly higher than the precision values of the other models, which range from 0.6907 to 0.9081. This suggests that DFL-ConvGRU

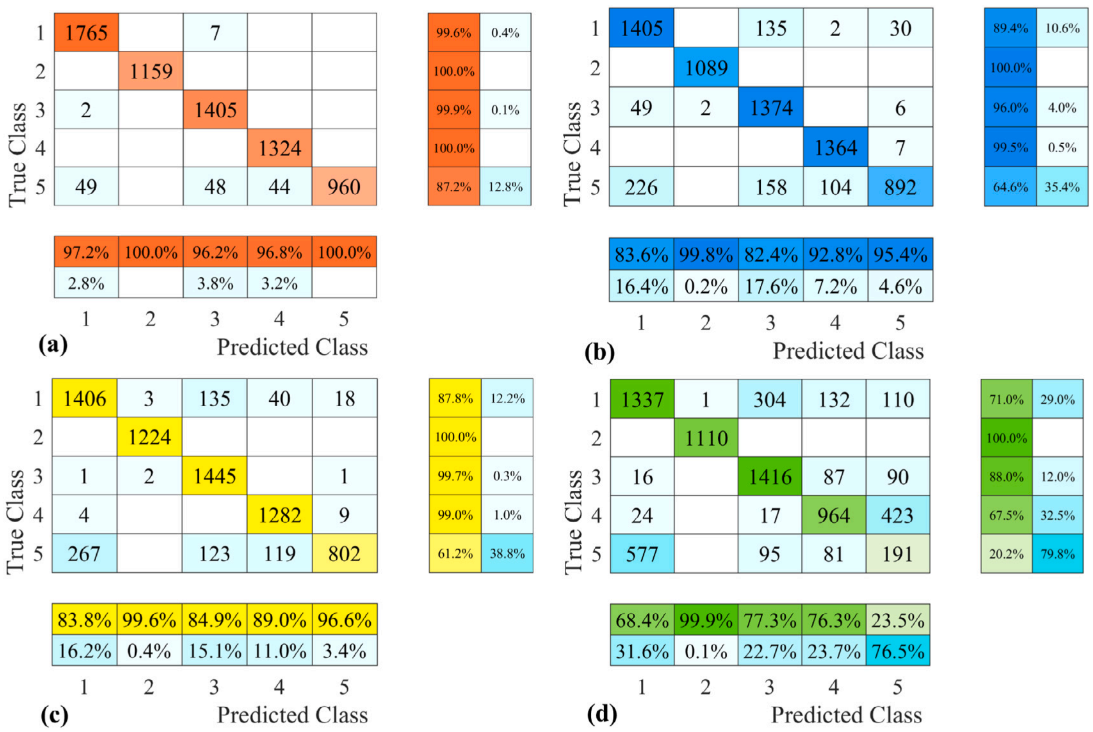

Fig. 11. Confusion matrix of competing techniques for known users (a) DFL-ConvGRU (b) DFL-GRU (c) DFL-LSTM (d) DFL-CNN.

Table 6
Comparison of theft detection for known users.

Tech.	Accuracy	Precision	Recall	F1Score
DFL-ConvGRU	0.9778	0.9804	0.9733	0.9758
DFL-GRU	0.8949	0.9081	0.8990	0.8962
DFL-LSTM	0.8951	0.9077	0.8953	0.8917
DFL-CNN	0.7194	0.6907	0.6934	0.6906

Table 7
Comparison of theft detection for unknown users across five theft types.

Tech.	Accuracy	Precision	Recall	F1Score
DFL-ConvGRU	0.9784	0.9679	0.9700	0.9717
DFL-GRU	0.9493	0.9203	0.8959	0.9066
DFL-LSTM	0.9517	0.9181	0.9058	0.9114
DFL-CNN	0.8786	0.8113	0.7517	0.7716

minimizes false positive predictions, providing a higher level of confidence in identifying actual theft cases. The recall value of 0.9733 for DFL-ConvGRU also demonstrates its ability to capture a large proportion of true positive theft instances.

Additionally, the F1 score of DFL-ConvGRU is 0.9758, which is consistently higher than the F1 scores of the other models. The F1 score considers both precision and recall, providing a balanced evaluation of a model's performance. The higher F1 score of DFL-ConvGRU implies a better trade-off between precision and recall, indicating a more robust and accurate theft detection capability.

5.3. Detection comparison for unknown users

In Table 7, DFL-ConvGRU achieves an accuracy of 97.84%, which is 9.98% higher than the lowest-performing model, DFL-CNN, with an accuracy of 87.86%. This indicates a substantial improvement in accurately classifying theft incidents. Moreover, DFL-ConvGRU achieves a precision of 96.79%, which is 7.66% higher than DFL-CNN, and a recall of 97.00%, surpassing the other models by at least 7.43%. The F1 score of DFL-ConvGRU, at 97.17%, demonstrates a 28.11% improvement over DFL-CNN. These percentages highlight the significant performance advantage of DFL-ConvGRU in theft detection.

Moving to Table 8, DFL-ConvGRU maintains its superior performance. With an accuracy of 94.38%, it outperforms the least accurate model, DFL-CNN, by 6.24%. DFL-ConvGRU achieves a precision of 89.90%, surpassing the other models by 11.78% or more. Similarly, DFL-ConvGRU's recall of 88.92% and F1 score of 89.30% exhibit improvements of at least 13.75% and 15.84% respectively compared to the lowest-performing model, DFL-CNN. These percentages illustrate the significant performance gap in favor of DFL-ConvGRU.

Table 8
Unknown users theft detection comparison with 6 theft types.

Tech.	Accuracy	Precision	Recall	F1Score
DFL-ConvGRU	0.9438	0.8990	0.8892	0.8930
DFL-GRU	0.8737	0.7697	0.7617	0.7641
DFL-LSTM	0.8707	0.7474	0.7585	0.7433
DFL-CNN	0.8122	0.7812	0.7401	0.7346

Finally, in Table 9, DFL-ConvGRU remains the top-performing model. It achieves an accuracy of 88.46%, outperforming DFL-CNN by 12.60%. DFL-ConvGRU's precision of 84.10% surpasses the other models by at least 20.06%, while its recall of 83.18% exceeds the lowest-performing model, DFL-CNN, by 24.95%. The F1 score of DFL-ConvGRU, at 84.34%, represents a 30.33% improvement over DFL-CNN. These percentages highlight the substantial performance advantage of DFL-ConvGRU in accurately identifying theft incidents.

A comprehensive examination unequivocally illustrates the enhanced efficacy of DFL-ConvGRU in detecting theft. Consistently outperforming other models, DFL-ConvGRU consistently attains notably elevated percentages in accuracy, precision, recall, and F1 score, accentuating its proficiency in precisely pinpointing theft occurrences. These findings underscore DFL-ConvGRU's supremacy and underscore its viability as a dependable solution for theft detection.

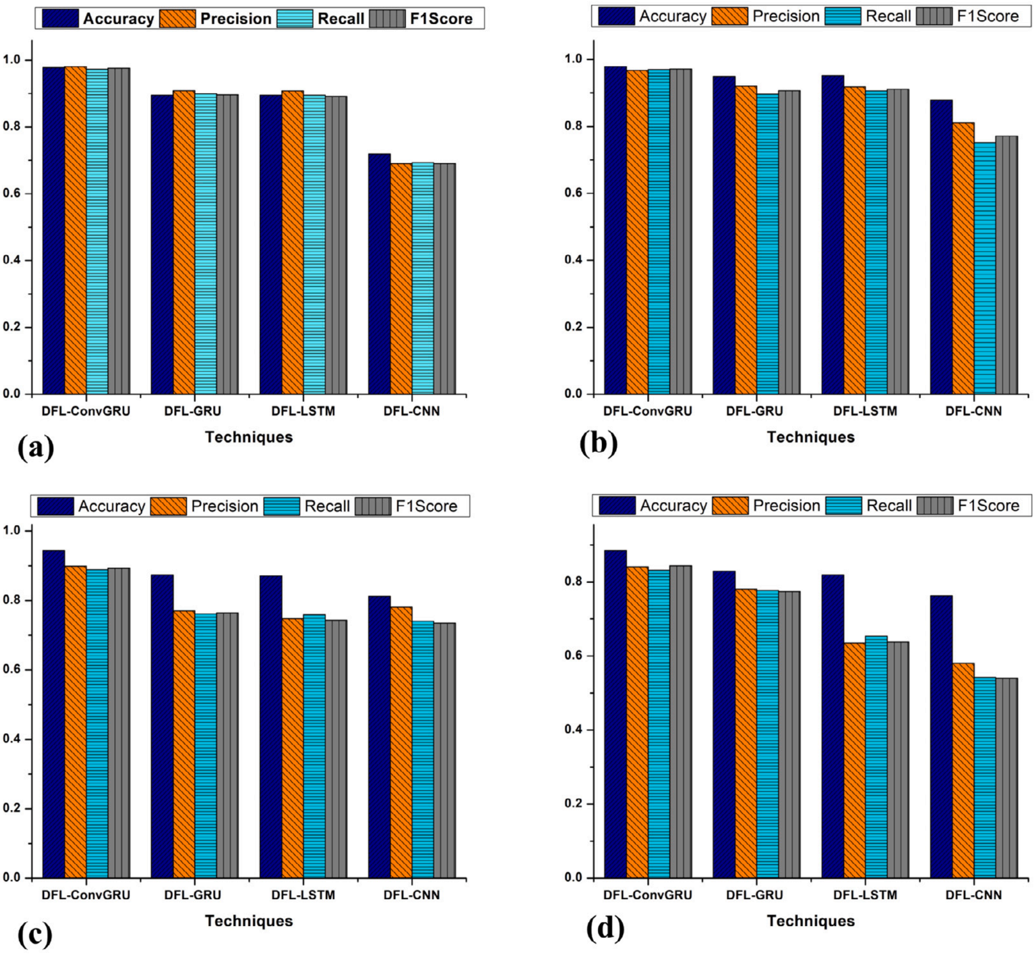

Fig. 12. Comparison of techniques for (a) Known Users (b) Un-Known Users with 5 Thefts (c) Un-Known Users with 6 Thefts (d) Un-Known Users with 7 Thefts.

Table 9
Comparison of theft detection for unknown users across seven theft types.

Tech.	Accuracy	Precision	Recall	F1Score
DFL-ConvGRU	0.8846	0.8410	0.8318	0.8434
DFL-GRU	0.8290	0.7803	0.7767	0.7738
DFL-LSTM	0.8192	0.6350	0.6539	0.6377
DFL-CNN	0.7620	0.5804	0.5423	0.5401

5.4. Comparative analysis

The comparative analysis presents a detailed evaluation of theft detection techniques, including state-of-the-art (SOTA) models. Our model, DFL-ConvGRU, achieves a remarkable accuracy of 97.84%, outperforming the other SOTA models listed in Table 10. For instance, the Feed Forward DNN achieves an accuracy of 91.8%, while the AlexNet-AdaBoost-ABC model achieves 88% accuracy. The ETD-ConvLSTM model achieves 96.3% accuracy, and the DAL-CNN model achieves 95.1% accuracy. Furthermore, the HRS-WSVDD model achieves an accuracy of 96.8%. Our model's exceptional accuracy of 97.84% clearly demonstrates its superiority over existing techniques. These results highlight the significant advancements offered by DFL-ConvGRU in the field of theft detection, solidifying its position as a highly effective and reliable solution for accurately identifying theft incidents.

5.5. Discussion

The comprehensive analysis of the results from the provided tables showcases the superior performance of DFL-ConvGRU in theft detection. In diverse scenarios encompassing both familiar and unfamiliar users, alongside varying counts of theft types, DFL-ConvGRU consistently exhibited superior performance compared to alternative models, as evidenced by its consistently higher accuracy, precision, recall, and F1 score. This indicates that DFL-ConvGRU is a robust and reliable solution for accurately identifying theft incidents.

The success of DFL-ConvGRU can be attributed to its unique architecture, which combines the strengths of CNN and gated GRU. The CNN component enables the model to effectively extract spatial features from the input data, capturing intricate patterns associated with theft. The GRU component, on the other hand, captures temporal dependencies and long-term relationships within the sequence, enhancing the model's ability to understand the sequential nature of theft incidents. By leveraging both spatial and temporal information, DFL-ConvGRU achieves a more comprehensive understanding of theft patterns, leading to its superior performance.

The comparative analysis also highlighted the limitations of other models, such as DFL-GRU, DFL-LSTM, and DFL-CNN, in accurately detecting theft incidents. These models exhibited lower accuracy and other performance metrics compared to DFL-ConvGRU. The results underline the importance of incorporating both spatial and temporal information, as achieved by DFL-ConvGRU, for effective theft detection.

Table 10
Comparative analysis with State-of-the-Art (SOTA) techniques.

Ref.	Year	Tech.	Accuracy
Lepolesa et al. (2022)	2022	Feed Forward DNN	0.918
Ullah et al. (2022)	2022	AlexNet-AdaBoost-ABC	0.880
Xia et al. (2023)	2023	ETD-ConvLSTM	0.963
Zhu et al. (2023)	2023	DAL-CNN	0.951
Cai et al. (2023)	2023	HRS-WSVDD	0.968
Our Model	2023	DFL-ConvGRU	0.9784

Therefore, the analysis of the results firmly establishes DFL-ConvGRU as the top-performing model in theft detection, surpassing other state-of-the-art techniques. Its robust performance, driven by the integration of CNN and GRU, provides a valuable contribution to the field of theft detection. The findings from this study have implications for researchers and practitioners, emphasizing the significance of considering both spatial and temporal aspects in developing accurate and reliable theft detection systems.

5.5.1. Alignment with unified Smart Grid and industry 5.0 concepts

Smart grid in recent decades has been equipped with digital sensing, control and connectivity across the board. The monitoring specially is done via IP that is a basic part of industry 5.0 where the artificial intelligence plays a significant role in diagnostics of control accuracy and anomaly detection. These properties of modern smart grid allow for the federated learning to be effective. The band width of communication channels allows for the throughput capable enough for the Federated Learning to be effective in this scenario. Smart Grid in industrial applications confines to these standards. IEEE standards often play a significant role in shaping communication protocols for the Smart Grid i.e. IEC 61850. This is a widely adopted international standard for the design of substation automation and communication systems. It defines a set of communication profiles for electric substations and is essential for the integration of intelligent electronic devices (IEDs) in substations. Similarly, IEC 62056 combination of DLMS (Device Language Message Specification) and COSEM (Companion Specification for Energy Metering) are used for meter reading and data exchange between various devices in the grid, including smart meters and data concentrators. From the security point of view IEC 62351 is regarded well in smart grids. This series of standards addresses the security of industrial automation and control systems, including those used in the Smart Grid. It provides guidelines for securing communication and data exchange. The models generated by the FL technique are well handled by the smart grid constraints. Authors estimate the Smart grid connected VIA WIFI 6802.11n operating at 2.4 GHz delivers 450 Mbps which is adequate for real time theft detection. The additional benefits of alignment of proposed technique with Smart grid in Industry 5.0 can be listed as:

- Enhanced Energy Management: The newly proposed Convolutional Gated Recurrent Unit (ConvGRU) model aligns with the unified Smart Grid and Industry 5.0 concept by significantly improving energy management. In Industry 5.0, systems are expected to be highly responsive and adaptable. The ConvGRU model captures not only temporal patterns but also spatial patterns in electricity consumption data, enabling real-time adjustments in energy distribution. This enhances the Smart Grid's ability to respond dynamically to changes in energy demand and supply.
- Data-Driven Decision Making: Both Industry 5.0 and the unified Smart Grid emphasize data-driven decision making. The ConvGRU model leverages deep federated learning to analyze distributed data sources. By doing so, it facilitates data-driven insights into electricity consumption patterns and identifies potential anomalies, such as theft. This aligns perfectly with Industry 5.0's focus on intelligent data analysis for making informed decisions.
- Privacy-Preserving Collaboration: Industry 5.0 promotes collaborative ecosystems where various stakeholders work together efficiently. The federated learning aspect of the proposed model allows data sharing and model training while maintaining data privacy. This aligns with the Smart Grid's need for utilities, consumers, and regulators to collaborate securely and transparently, fostering a trusted energy ecosystem.
- Resilience and Security: Industry 5.0 seeks to enhance the resilience of industrial systems. The theft detection technique improves the security of Smart Grid networks by detecting unauthorized activities, thus contributing to the resilience of the grid against external threats. In a unified Smart Grid, where various interconnected components interact, security becomes paramount.
- Consumer-Centric Approach: Industry 5.0 emphasizes a more customer-centric approach to industry. The proposed technique indirectly benefits consumers by helping to reduce electricity theft, which can lead to lower energy costs and more reliable service. Smart Grids, when secure and efficient, can provide consumers with greater control and options in managing their energy consumption.

5.5.2. Application-oriented benefits

The application-oriented benefits for the proposed model can be from real time fraud detection, load balancing by adjusting to the demand of the consumers, optimization of physical infrastructure resources instead of standardizing the equipment saving costs of implementation and engaging the customers by minimizing the transmission losses. Since it ultimately impacts the cost of electricity hence minimizing the costs and theft will improvise cost/watt saving for the customers. These application-oriented benefits can be listed as follow:

- Real-Time Fraud Detection: The ConvGRU model's ability to detect electricity theft in real-time aligns with Industry 5.0's requirement for immediate responses to anomalies. This has direct applications in minimizing revenue losses for utilities and ensuring fair billing for consumers.
- Dynamic Load Balancing: By analyzing consumption patterns, the proposed model enables dynamic load balancing. Utilities can optimize energy distribution, reduce peak loads, and prevent outages. In Industry 5.0, such capabilities are crucial for maintaining uninterrupted operations.
- Infrastructure Optimization: With accurate theft detection and insights into consumption, utilities can optimize infrastructure investments. They can prioritize upgrades and maintenance based on actual usage patterns, thus reducing unnecessary spending.
- Regulatory Compliance: Smart Grids must adhere to various regulations. The proposed model aids in compliance by ensuring fair and transparent operations, which is essential in the context of Industry 5.0's focus on regulatory alignment.
- Customer Engagement: The theft detection technique can be part of customer engagement strategies. By demonstrating a commitment to security and fair billing, utilities can improve customer satisfaction and loyalty.

6. Conclusion

The integration of Smart Grid technology and Industry 5.0 concepts has driven the need for advanced energy management systems, but the threat of electricity theft poses a significant challenge to their security and reliability. This paper presented a secure and reliable theft detection technique for Smart Grid networks using deep federated learning (FL) and a Convolutional Gated Recurrent Unit (ConvGRU) model. The proposed technique leverages FL to train a ConvGRU model on distributed data sources while preserving data privacy.

The results of the study demonstrate the efficacy of the deep FL-based ConvGRU model in accurately detecting electricity theft. The comparative analysis showcases the superior performance of the proposed technique, as reflected in the high accuracy, precision, recall, and F1 score achieved. These results validate the effectiveness of combining convolutional and gated recurrent units to capture both spatial and temporal patterns in electricity consumption data.

The research contributes to the enhancement of security and efficiency in Smart Grid systems. By utilizing FL, the technique ensures that data remains distributed and private, addressing concerns regarding data privacy. The findings highlight the potential of the proposed approach in combating electricity theft, thus bolstering the security and reliability of Smart Grid networks.

To further advance theft detection in the Smart Grid, future research can focus on hyperparameter optimization and explore alternative deep-learning architectures. These avenues of investigation aim to improve the technique's performance and refine its ability to accurately identify theft incidents. Ultimately, the proposed technique and its promising results pave the way for the development of robust and secure systems in the Smart Grid domain.

CRediT authorship contribution statement

Muhammad Hamza Zafar: Conceptualization, Methodology, Resources, Project administration. **Syed Muhammad Salman Bukhari:** Conceptualization, Methodology, Resources, Project administration. **Mohamad Abou Houran:** Validation, Data curation. **Syed Kumayl Raza Moosavi:** Conceptualization, Formal analysis, Investigation. **Majad Mansoor:** Visualization, Data curation. **Nedaa Al-Tawalbeh:** Validation, Formal analysis. **Filippo Sanfilippo:** Supervision, Funding acquisition, Investigation.

Declaration of competing interest

None. All authors claim that there is not any conflict of interest regarding the above submission. The work of this submission has not been published previously. It is not under consideration for publication elsewhere. Its publication is approved by all authors and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright-holder.

Data availability

The link is provided to the data used in this work.

Acknowledgments

This research is supported by the Top Research Center Mechatronics (TRCM), University of Agder (UiA), Norway.

References

Abou Houran, M., Bukhari, S.M.S., Zafar, M.H., Mansoor, M., Chen, W., 2023. COA-CNN-LSTM: Coati optimization algorithm-based hybrid deep learning model for PV/wind power forecasting in smart grid applications. Appl. Energy 349, 121638.

Arango, L.G., Deccache, E., Bonatto, B.-H.D., Arango, H., Pamplona, E.O., 2017. Study of electricity theft impact on the economy of a regulated electricity company. J. Control, Autom. Electr. Syst. 28 (4), 567–575.

Ashraf, M.M., Waqas, M., Abbas, G., Baker, T., Abbas, Z.H., Alasmary, H., 2022. Feddp: A privacy-protecting theft detection scheme in smart grids using federated learning. Energies 15 (17), 6241.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization.. J. Mach. Learn. Res. 13 (2).

Bohani, F.A., Suliman, A., Saripuddin, M., Sameon, S.S., Md Salleh, N., Nazeri, S., Mandeep, J.S., 2021. A comprehensive analysis of supervised learning techniques for electricity theft detection. J. Electr. Comput. Eng. 2021, 1–10.

Cai, Q., Li, P., Wang, R., 2023. Electricity theft detection based on hybrid random forest and weighted support vector data description. Int. J. Electr. Power Energy Syst. 153, 109283.

Cao, Y., Li, Q., Tan, Y., Li, Y., Chen, Y., Shao, X., Zou, Y., 2018. A comprehensive review of energy internet: basic concept, operation and planning methods, and research prospects. J. Mod. Power Syst. Clean Energy 6 (3), 399–411.

Depuru, S.S.S.R., Wang, L., Devabhaktuni, V., 2011. Electricity theft: Overview, issues, prevention and a smart meter based approach to control theft. Energy Policy 39 (2), 1007–1015.

Eggensperger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H., Leyton-Brown, K., et al., 2013. Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In: NIPS Workshop on Bayesian Optimization in Theory and Practice, Vol. 10. (3).

Elayan, H., Aloqaily, M., Guizani, M., 2021. Sustainability of healthcare data analysis IoT-based systems using deep federated learning. IEEE Internet Things J. 9 (10), 7338–7346.

European Commission, Directorate-General for Research and Innovation, 2021. Industry 5.0: Towards a Sustainable, Human-centric and Resilient European Industry. Publications Office of the European Union.

Fang, X., Misra, S., Xue, G., Yang, D., 2011. Smart grid—The new and improved power grid: A survey. IEEE Commun. Surv. Tutor. 14 (4), 944–980.

Gholinejad, H.R., Loni, A., Adabi, J., Marzband, M., 2020. A hierarchical energy management system for multiple home energy hubs in neighborhood grids. J. Build. Eng. 28, 101028.

Gul, H., Javaid, N., Ullah, I., Qamar, A.M., Afzal, M.K., Joshi, G.P., 2020. Detection of non-technical losses using sostlink and bidirectional gated recurrent unit to secure smart meters. Appl. Sci. 10 (9), 3151.

Haq, E.U., Pei, C., Zhang, R., Jianjun, H., Ahmad, F., 2023. Electricity-theft detection for smart grid security using smart meter data: A deep-CNN based approach. Energy Rep. 9, 634–643, 2022 9th International Conference on Power and Energy Systems Engineering.

Hasan, M.N., Toma, R.N., Nahid, A.-A., Islam, M.M., Kim, J.-M., 2019. Electricity theft detection in smart grid systems: A CNN-LSTM based approach. Energies 12 (17), 3310.

Henriques, H.O., Corrêa, R.L.S., Fortes, M.Z., Borba, B.S.M.C., Ferreira, V.H., 2020. Monitoring technical losses to improve non-technical losses estimation and detection in LV distribution systems. Measurement 161, 107840.

Hussain, S., et al., 2021. A novel feature engineered-CatBoost-based supervised machine learning framework for electricity theft detection. Energy Rep. 7, 4425–4436.

Jadidbonab, M., Mohammadi-Ivatloo, B., Marzband, M., Siano, P., 2020. Short-term self-scheduling of virtual energy hub plant within thermal energy market. IEEE Trans. Ind. Electron. 68 (4), 3124–3136.

Jiang, R., Lu, R., Wang, Y., Luo, J., Shen, C., Shen, X., 2014. Energy-theft detection issues for advanced metering infrastructure in smart grid. Tsinghua Sci. Technol. 19 (2), 105–120.

Jokar, P., Arianpoo, N., Leung, V.C., 2015. Electricity theft detection in AMI using customers' consumption patterns. IEEE Trans. Smart Grid 7 (1), 216–226.

Kabalci, Y., 2016. A survey on smart metering and smart grid communication. Renew. Sustain. Energy Rev. 57, 302–318.

Karnouskos, S., Terzidis, O., Karnouskos, P., 2007. An advanced metering infrastructure for future energy networks. In: Labiod, H., Badra, M. (Eds.), New Technologies, Mobility and Security. Springer Netherlands, Dordrecht, pp. 597–606.

Khan, Z.A., Adil, M., Javaid, N., Saqib, M.N., Shafiq, M., Choi, J.-G., 2020. Electricity theft detection using supervised learning techniques on smart meter data. Sustainability 12 (19), 8023.

Khan, K., Zafar, M.H., Khan, N.M., Khan, U.A., 2021. Optimal control of PV system to extract maximum power under non-uniform environmental conditions. In: 2021 16th International Conference on Emerging Technologies. (ICET), IEEE, pp. 1–6.

Leite, J.B., Mantovani, J.R.S., 2016. Detecting and locating non-technical losses in modern distribution networks. IEEE Trans. Smart Grid 9 (2), 1023–1032.

Leng, J., Sha, W., Wang, B., Zheng, P., Zhuang, C., Liu, Q., Wuest, T., Mourtzis, D., Wang, L., 2022. Industry 5.0: Prospect and retrospect. J. Manuf. Syst. 65, 279–295.

Lepolesa, L.J., Achari, S., Cheng, L., 2022. Electricity theft detection in smart grids based on deep neural network. Ieee Access 10, 39638–39655.

Li, L., Fan, Y., Tse, M., Lin, K.-Y., 2020. A review of applications in federated learning. Comput. Ind. Eng. 149, 106854.

Li, S., Han, Y., Yao, X., Yingchen, S., Wang, J., Zhao, Q., 2019. Electricity theft detection in power grids with deep learning and random forests. J. Electr. Comput. Eng. 2019, 1–12.

Liao, W., Yang, Z., Liu, K., Zhang, B., Chen, X., Song, R., 2023. Electricity theft detection using euclidean and graph convolutional neural networks. IEEE Trans. Power Syst. 38 (4), 3514–3527.

Marzband, M., Azarinejadian, F., Savaghebi, M., Pouresmaeil, E., Guerrero, J.M., Lightbody, G., 2018. Smart transactive energy framework in grid-connected multiple home microgrids under independent and coalition operations. Renew. Energy 126, 95–106.

McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. PMLR, pp. 1273–1282.

Mian Qaisar, S., 2020. Event-driven coulomb counting for effective online approximation of li-ion battery state of charge. Energies 13 (21), 5600.

Mujeeb, S., Javaid, N., 2019. ESAENARX and DE-RELM: Novel schemes for big data predictive analytics of electricity load and price. Sustainable Cities Soc. 51, 101642.

Nagaraj, B., Malagi, K.B., 2023. Boosting the accuracy of optimisation chatbot by random forest with halving grid search hyperparameter tuning. ICTACT J. Soft Comput. 13 (3).

Ogunsanya, M., Isichei, J., Desai, S., 2023. Grid search hyperparameter tuning in additive manufacturing processes. Manuf. Lett..

Ozcanli, A.K., Baysal, M., 2022. Islanding detection in microgrid using deep learning based on 1D CNN and CNN-LSTM networks. Sustain. Energy, Grids Netw. 32, 100839.

Savian, F.d.S., Siluk, J.C.M., Garlet, T.B., do Nascimento, F.M., Pinheiro, J.R., Vale, Z., 2021. Non-technical losses: A systematic contemporary article review. Renew. Sustain. Energy Rev. 147, 111205.

Soper, D.S., 2022. Hyperparameter optimization using successive halving with greedy cross validation. Algorithms 16 (1), 17.

Subramanian, B., Olimov, B., Naik, S.M., Kim, S., Park, K.-H., Kim, J., 2022. An integrated mediapipe-optimized GRU model for Indian sign language recognition. Sci. Rep. 12 (1), 11964.

Sun, M., Wang, Y., Strbac, G., Kang, C., 2018. Probabilistic peak load estimation in smart cities using smart meter data. IEEE Trans. Ind. Electron. 66 (2), 1608–1618.

Ullah, A., Javaid, N., Asif, M., Javed, M.U., Yahaya, A.S., 2022. Alexnet, adaboost and artificial bee colony based hybrid model for electricity theft detection in smart grids. Ieee Access 10, 18681–18694.

Villalobos-Arias, L., Quesada-López, C., 2021. Comparative study of random search hyper-parameter tuning for software effort estimation. In: Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering. pp. 21–29.

Wang, S.-C., 2003. Artificial neural network. In: Wang, S.-C. (Ed.), Interdisciplinary Computing in Java Programming. Springer US, Boston, MA, pp. 81–100.

Wang, Y., Chen, Q., Kang, C., Xia, Q., 2016. Clustering of electricity consumption behavior dynamics toward big data applications. IEEE Trans. Smart Grid 7 (5), 2437–2447.

Wen, M., Xie, R., Lu, K., Wang, L., Zhang, K., 2022. FedDetect: A novel privacy-preserving federated learning framework for energy theft detection in smart grid. IEEE Internet Things J. 9 (8), 6069–6080.

Xia, X., Lin, J., Jia, Q., Wang, X., Ma, C., Cui, J., Liang, W., 2023. ETD-ConvLSTM: A deep learning approach for electricity theft detection in smart grids. IEEE Trans. Inf. Forensics Secur..

Xia, X., Xiao, Y., Liang, W., Cui, J., 2022. Detection methods in smart meters for electricity thefts: A survey. Proc. IEEE 110 (2), 273–319.

Yan, Z., Wen, H., 2021a. Electricity theft detection base on extreme gradient boosting in AMI. IEEE Trans. Instrum. Meas. 70, 1–9.

Yan, Z., Wen, H., 2021b. Performance analysis of electricity theft detection for the smart grid: An overview. IEEE Trans. Instrum. Meas. 71, 1–28.

Yang, Q., Liu, Y., Chen, T., Tong, Y., 2019. Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol. 10 (2), 1–19.

Yu, H., Liu, Z., Liu, Y., Chen, T., Cong, M., Weng, X., Niyato, D., Yang, Q., 2020. A sustainable incentive scheme for federated learning. IEEE Intell. Syst. 35 (4), 58–69.

Zafar, M.H., Mansoor, M., Abou Houran, M., Khan, N.M., Khan, K., Moosavi, S.K.R., Sanfilippo, F., 2023. Hybrid deep learning model for efficient state of charge estimation of li-ion batteries in electric vehicles. Energy 282, 128317.

Zeng, N., Song, D., Li, H., You, Y., Liu, Y., Alsaadi, F.E., 2021. A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution. Neurocomputing 432, 170–182.

Zhang, C., Li, S., Xia, J., Wang, W., Yan, F., Liu, Y., 2020. $\{\mathrm{BatchCrypt}\}$: Efficient homomorphic encryption for $\{\mathrm{Cross}-\mathrm{Silo}\}$ federated learning. In: 2020 USENIX Annual Technical Conference. (USENIX ATC 20), pp. 493–506.

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., Gao, Y., 2021. A survey on federated learning. Knowl.-Based Syst. 216, 106775.

Zheng, K., Chen, Q., Wang, Y., Kang, C., Xia, Q., 2018. A novel combined data-driven approach for electricity theft detection. IEEE Trans. Ind. Inform. 15 (3), 1809–1819.

Zhu, G., Du, Y., Gündüz, D., Huang, K., 2020. One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis. IEEE Trans. Wireless Commun. 20 (3), 2120–2135.

Zhu, L., Wen, W., Li, J., Zhang, C., Zhou, B., Shuai, Z., 2023. Deep active learning-enabled cost-effective electricity theft detection in smart grids. IEEE Trans. Ind. Inform..