

# 1. Phd\_thesis\_Chandan Kumar 2K19PhDEE505.pdf

 Delhi Technological University

---

## Document Details

### Submission ID

trn:oid:::27535:129047228

### Submission Date

Feb 24, 2026, 2:27 PM GMT+5:30

### Download Date

Feb 24, 2026, 2:33 PM GMT+5:30

### File Name

1. Phd\_thesis\_Chandan Kumar 2K19PhDEE505.pdf

### File Size

9.6 MB

96 Pages

20,412 Words

112,792 Characters

# 9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Filtered from the Report

- Bibliography
- Cited Text
- Small Matches (less than 13 words)
- Submitted works

## Exclusions

- 2 Excluded Sources
- 1 Excluded Match

## Match Groups

-  **63 Not Cited or Quoted 7%**  
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**  
Matches that are still very similar to source material
-  **18 Missing Citation 2%**  
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**  
Matches with in-text citation present, but no quotation marks

## Top Sources

- 7%  Internet sources
- 8%  Publications
- 0%  Submitted works (Student Papers)

## Integrity Flags

### 1 Integrity Flag for Review

-  **Replaced Characters**  
244 suspect characters on 16 pages  
Letters are swapped with similar characters from another alphabet.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Match Groups

- 63 Not Cited or Quoted 7%**  
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**  
Matches that are still very similar to source material
- 18 Missing Citation 2%**  
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**  
Matches with in-text citation present, but no quotation marks

## Top Sources

- 7% Internet sources
- 8% Publications
- 0% Submitted works (Student Papers)

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

|    |             |                                                                                      |     |
|----|-------------|--------------------------------------------------------------------------------------|-----|
| 1  | Internet    | pdfs.semanticscholar.org                                                             | 2%  |
| 2  | Internet    | www.itu.int                                                                          | <1% |
| 3  | Publication | Lei Che, Xiaoping Yang, Liang Wang. "Text Feature Extraction Based on Stacked V...   | <1% |
| 4  | Internet    | aclweb.org                                                                           | <1% |
| 5  | Internet    | www.mdpi.com                                                                         | <1% |
| 6  | Publication | Kisung Park, Joonyoung Lee, Ashok Kumar Das, Youngho Park. "BPPS:Blockchain-...      | <1% |
| 7  | Publication | Randhir Kumar, Ahamed Aljuhani, Danish Javeed, Prabhat Kumar, Shareeful Isla...      | <1% |
| 8  | Publication | Yifan Huang, Qifeng Xu. "Electricity theft detection based on stacked sparse deno... | <1% |
| 9  | Publication | Prabhat Kumar, Govind P. Gupta, Rakesh Tripathi. "An ensemble learning and fog...    | <1% |
| 10 | Internet    | lutpub.lut.fi                                                                        | <1% |

|    |             |                                                                                      |     |
|----|-------------|--------------------------------------------------------------------------------------|-----|
| 11 | Internet    | uobrep.openrepository.com                                                            | <1% |
| 12 | Internet    | www.biorxiv.org                                                                      | <1% |
| 13 | Publication | Randhir Kumar, Prabhat Kumar, Rakesh Tripathi, Govind P. Gupta, Neeraj Kumar,...     | <1% |
| 14 | Internet    | downloads.hindawi.com                                                                | <1% |
| 15 | Internet    | espace.etsmtl.ca                                                                     | <1% |
| 16 | Publication | Randhir Kumar, Prabhat Kumar, Rakesh Tripathi, Govind P. Gupta, Sahil Garg, Mo...    | <1% |
| 17 | Publication | Abdallah R. Gad, Ahmed A. Nashat, Tamer M. Barkat. "Intrusion Detection System...    | <1% |
| 18 | Publication | Prabhat Kumar, Randhir Kumar, Govind P. Gupta, Rakesh Tripathi, Alireza Jolfaei, ... | <1% |
| 19 | Publication | Prabhat Kumar, Randhir Kumar, Gautam Srivastava, Govind P. Gupta, Rakesh Trip...     | <1% |
| 20 | Internet    | ieeexplore.ieee.org                                                                  | <1% |
| 21 | Publication | Yifan Huang, Zhengnan Yao, Qifeng Xu. "Classification model of electricity consu...  | <1% |
| 22 | Internet    | www.researchgate.net                                                                 | <1% |
| 23 | Internet    | www.coursehero.com                                                                   | <1% |
| 24 | Publication | Abdallah R. Gad, Mohamed Haggag, Ahmed A. Nashat, Tamer M. Barakat. "A Dist...       | <1% |

|    |             |                                                                                      |     |
|----|-------------|--------------------------------------------------------------------------------------|-----|
| 25 | Internet    | scm.etu.ru                                                                           | <1% |
| 26 | Internet    | www.vskills.in                                                                       | <1% |
| 27 | Publication | Ahamed Aljuhani, Prabhat Kumar, Rehab Alanazi, Turki Albalawi, Okba Taouali, A....   | <1% |
| 28 | Publication | Charithri Yapa, Chamitha de Alwis, Madhusanka Liyanage, Janaka Ekanayake. "Su...     | <1% |
| 29 | Publication | Randhir Kumar, Prabhat Kumar, Rakesh Tripathi, Govind P. Gupta, A. K. M. Najmu...    | <1% |
| 30 | Publication | J. Kalidass, T. Purusothaman, P. Suresh. "Enhancement of end-to-end security in a... | <1% |
| 31 | Internet    | arxiv.org                                                                            | <1% |
| 32 | Publication | Randhir Kumar, Prabhat Kumar, Rakesh Tripathi, Govind P. Gupta, Thippa Reddy ...     | <1% |
| 33 | Internet    | ijsrcse.isroset.org                                                                  | <1% |
| 34 | Internet    | link.springer.com                                                                    | <1% |
| 35 | Publication | Prabhat Kumar, Randhir Kumar, Abhinav Kumar, A. Antony Franklin, Alireza Jolfa...    | <1% |
| 36 | Publication | Randhir Kumar, Prabhat Kumar, Abhinav Kumar, A. Antony Franklin, Alireza Jolfa...    | <1% |
| 37 | Internet    | ebin.pub                                                                             | <1% |
| 38 | Internet    | openaccess.city.ac.uk                                                                | <1% |

|    |             |                                                                                 |     |
|----|-------------|---------------------------------------------------------------------------------|-----|
| 39 | Internet    | www.frontiersin.org                                                             | <1% |
| 40 | Internet    | export.arxiv.org                                                                | <1% |
| 41 | Internet    | www.intechopen.com                                                              | <1% |
| 42 | Publication | "Medical Image Computing and Computer-Assisted Intervention – MICCAI 2017", ... | <1% |
| 43 | Internet    | www.phdinfo.org                                                                 | <1% |

# Chapter 1

## Introduction

Electricity has become one of the most vital utilities in modern society, powering homes, industries, healthcare services, communication systems, and transportation networks. Without reliable access to electricity, daily life would come to a standstill, and essential services would face severe disruption. Over the past decade, traditional power grids have evolved into what is now called the Smart Grid (SG), an advanced energy network that integrates information and communication technologies with electricity generation, transmission, and distribution. The purpose of this modernization is to make power systems more efficient, reliable, sustainable, and interactive. However, with these technological advancements come new challenges, especially in the domain of cybersecurity and overall system protection.

An SG differs significantly from the traditional grid in terms of design and operation. In the traditional grid model, electricity flowed in a one-way direction from large centralized plants to consumers. In contrast, the SG is a two-way communication system that allows not only the delivery of power but also the sharing of information between providers and consumers. Smart Meters (SM), automated distribution, and integration of renewable energy like wind and solar are some of the features that make smart grids highly efficient. Nevertheless, this increase in

efficiency is accompanied by a parallel increase in complexity, which broadens the attack surface for potential threats. Devices, networks, and control systems are all interconnected, meaning any weak link can be exploited by malicious actors to cause large-scale damage.

Cybersecurity for SG is not just a technical matter; it is also a matter of national security and public trust. Power grids are categorized as critical infrastructure, and any disruption can cascade into failures across multiple sectors. For example, a cyberattack on the power grid could interrupt healthcare services, shut down transportation systems, or trigger economic instability. Simple issues such as data breaches or false data injection attacks can lead to wrong decisions by grid operators, ultimately causing blackouts or equipment damage. Therefore, the need for a comprehensive and adaptable security framework is more urgent than ever.

### 1.1 Smart Grid Conceptual Model

The National Institute of Standards and Technology (NIST) defines the SG as an improved electricity network that combines sensing, communication, control, and automated decision-making with traditional energy infrastructure.

The NIST model, as can be seen in Figure 1.1, splits the SG into seven areas: Customers, Markets, Service Providers, Operations, Bulk Generation (which includes Distributed Energy Resources), Transmission, and Distribution. Each domain talks to each other through secure communication and electrical flows to make sure that electricity is delivered quickly and reliably.

Customer: They are the people who utilize the energy, store it, make it, and manage it.

Markets: A place where people and businesses buy and sell energy.

Service Providers: Companies that offer services to utilities and their customers.

Bulk Generation: This includes big generators and energy resources that are spread out.

Transmission: The long-distance transportation of bulk power.

Distribution: Sends and receives electricity to and from customers.

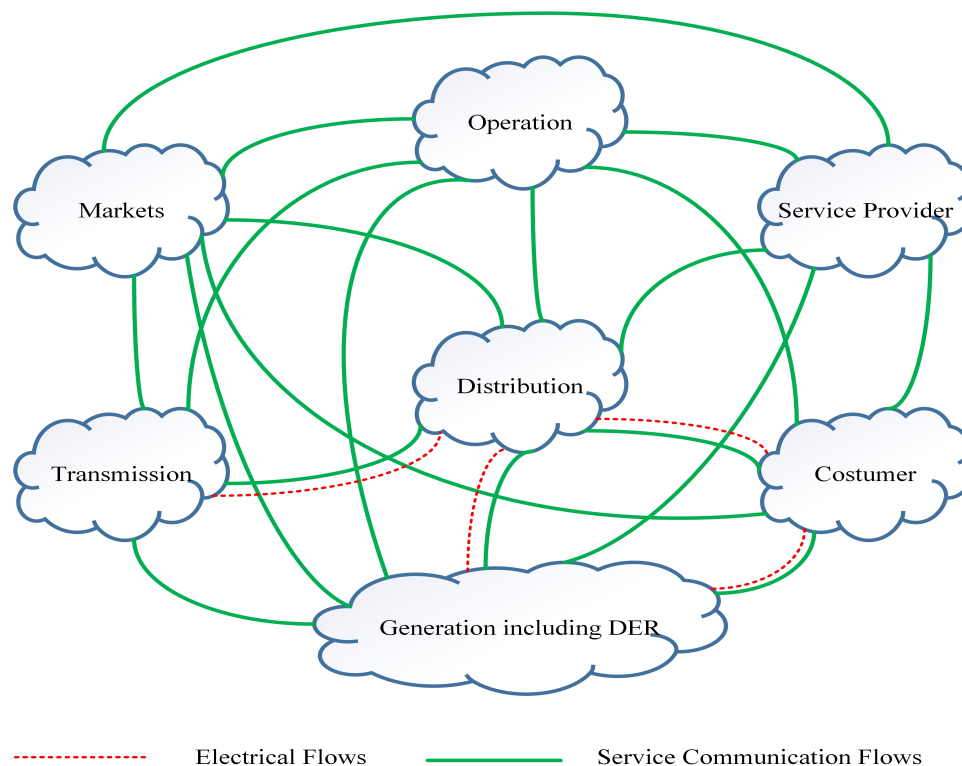


Figure 1.1: Smart Grid Conceptual Model (Source: DRAFT NIST SG FRAMEWORK[1])

## 1.2 Smart Grid 2.0 Architecture

The SG 2.0 set up is shown in Figure 1.2 There are mainly four layers, which are.

- 1) Physical Component Layer
- 2) Communication and Control Layer
- 3) Application Layer
- 4) Data Analysis Layer

**The physical component layer** has sensors that let you collect data for real-time monitoring and making smart decisions. Some examples are smart generation technologies, smart loads, smart sensors, SM, Phasor Measurement Units (PMU), Remote Terminal Units (RTU), Current Transformers (CT), and Voltage Transformers (VT). The exchange of collected data and information for real-time monitoring and management of SG 2.0 is made possible by communication infrastructure and internet-based protocols [2].

**The Communication and Control layer** makes sure that devices can talk to each other quickly, reliably, and in real time, with as little help from third-party service providers as possible.

**The application layer** of future SG will have new features, including microgeneration and large-scale power plants that use renewable energy sources, battery storage, and Electric Vehicles (EVs) trading with charging stations through peer-to-peer (P2P) energy-financial transactions. It also enables automated, efficient, and dependable transmission and distribution networks that allow energy to flow in both directions, as well as smart buildings and houses with SM that help manage load better [3].

The SG 2.0 infrastructure needs to be able to handle data, send data securely, safeguard privacy, and store data reliably [4, 5]. **The data analysis layer** helps with managing data in the cloud, using analytics to predict trends, and general control of the grid.

### 1.3 Security and Privacy Considerations

This section addresses the diverse privacy and security concerns surrounding smart grid infrastructure and explores how blockchain technology can effectively mitigate

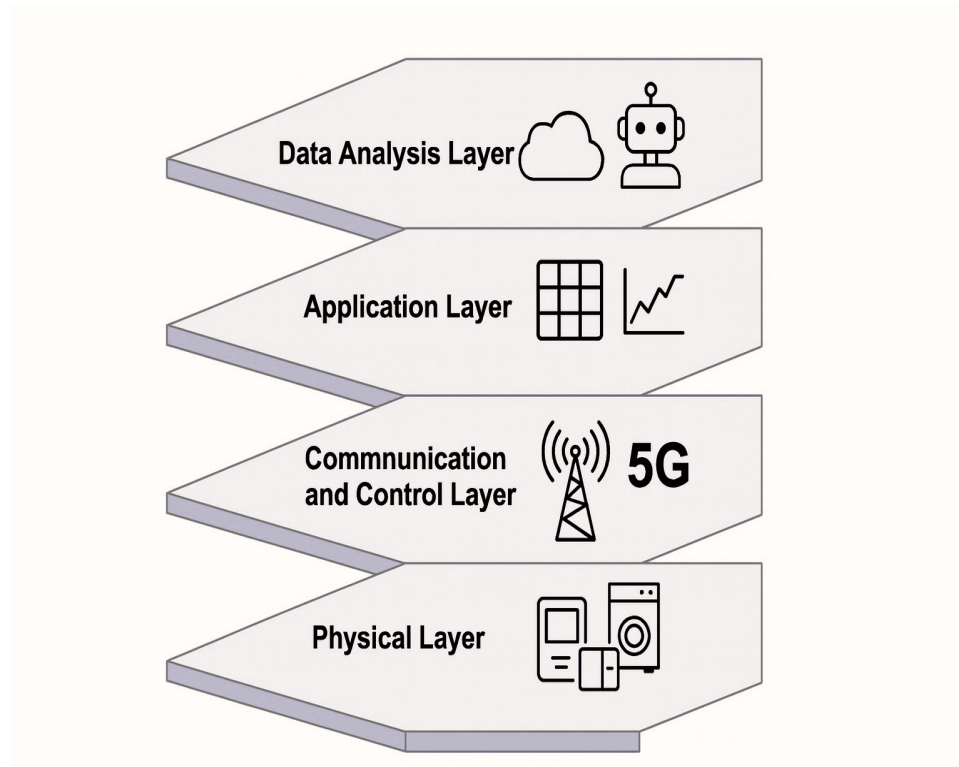


Figure 1.2: Smart Grid 2.0 Architecture

these challenges.

### 1.3.1 Threats in Smart Grids

Threats existing in smart grid can be categorised as

Physical Threats

Software Threats

Network Layer Threats

Control Layer Threats

Encryption Threats

**Physical Threats:** Physical threats on SG involve tampering with Sensor Nodes (SN) and SM to access confidential information. Other threats include jamming wireless sensor networks to disrupt communication, injecting malicious nodes or code to in-

tercept data, and False Data Injection attacks (FDI) that can affect the grid's state estimation and overall operation [6-8].

Software Threats: Software threats on smart grids include phishing, which compromises node authenticity, and malicious software like viruses that leads to data theft and tampering. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks disrupt communication between controllers and nodes, hindering grid operations [8, 9].

Network Threats: Predominantly include traffic analysis attacks, which give access to confidential information, and unauthorized access to data, which compromises the authenticity and confidentiality of the participating entities [8].

Control Layer Threats: Controller-related threats include (DoS) attacks, (FDI), and the insertion of malicious control algorithms or scripts, all of which disrupt the stable operation of the network. Additional threats involve unauthorized access to the Energy Management System (EMS) to undermine control functions and data cloning, further compromising the system's integrity[8].

Encryption-related threats: Encryption-related threats, such as cryptanalysis, threaten the confidentiality of information exchanged within SG and present a significant challenge to maintaining secure communication and data privacy in these advanced energy systems.

## 1.4 Blockchain and Smart Grid Integration

All levels of SG 2.0 use blockchain technology to maintain their decentralized structure. The smart grid needs a system that is open, P2P, and networked, with shared databases that encourage fairness and openness. It does away with single governance, adds new stakeholders and services, and works on its own to make things

more reliable and safe. The following parameters are explained in detail.

#### **1.4.1 Decentralized Architecture**

The increasing integration of distributed generation systems, such as renewable energy sources, EVs, and battery storage systems, poses challenges for centralized control due to their extensive geographical distribution and restricted manageability by a singular centralized authority. Blockchain technology is based on a decentralized paradigm, which replaces old ways of building trust with cryptographic proof to make processes safe and clear [10].

#### **1.4.2 Open structure**

Entities that meet the conditions for connecting to the grid can be allowed to take part in the flow of energy and information. Everyone who is linked to a public blockchain may see all of its nodes. Each participant has a copy of the distributed ledger, which makes sure that everything is as clear as possible. Private blockchains, on the other hand, only let authorized nodes see the data, which lets approved participants share data in a controlled way.

#### **1.4.3 Autonomous**

To assist in establishing a self-managing grid, Distributed Energy Resources (DERs), including renewable energy sources and storage devices, should be easy to link in a flexible, plug-and-play method. We need automated systems that balance the supply and demand for energy, find and fix problems, and operate the grid, all without much help from a central authority. Smart contracts are computer programs that run automatically when everyone on the blockchain agrees to certain circumstances

[11, 12].

#### 1.4.4 Interconnected

SG 2.0 moves away from the usual top-down approach and sets up a decentralized and distributed grid structure. Users in this network can also make energy, and every node can link to other nodes and the utility grid. This structure lets people trade and share information in a way that is open and can't be changed later [13, 14].

#### 1.4.5 Shared databases

To work on its own, a grid has to securely transmit up-to-date information on power prices, energy needs, and real-time measurements. Only people who have permission should be able to see this information. Blockchain is a system that lets approved members share a record of transactions that are structured in order [15, 16].

#### 1.4.6 Transparent

Transactions in the energy grid should be trackable, with information shared openly, securely, and in a way that cannot be changed afterward. Blockchain ensures transparency while protecting identities, using a shared digital ledger and unique codes for each participant. [17, 18].

#### 1.4.7 Equality

SG has transformed consumers into prosumers, allowing them to choose the energy sources they prefer to pay for. This creates a fair environment where all participants in energy trading have equal opportunities. It is important to verify the identity of users and allow grid access only to trusted stakeholders who meet the minimum requirements [19].

#### **1.4.8 Loss of Governness (Or Loss of Total Ownership of Network)**

No one person or group should have complete control over the authority. This method keeps users' private information safe (such as their name, energy use, and location) and gets rid of the chance of a single point of failure. Everyone who is allowed to participate has a copy of the transaction ledger, but changes may only be made if everyone agrees. Blockchain works on its own with distributed authority, which makes sure that data is safe, reliable, and private for users[20].

#### **1.4.9 New Stakeholders**

The system connects diverse energy sources with differing capacities at different levels of the power grid. Blockchain technology employs cryptographic methods, including Public Key Infrastructure (PKI), cryptographic hashing, and digital signatures to make secure connections and let people do business with each other without having to trust each other or have an intermediary check their work [21].

#### **1.4.10 New Services**

Microgrids that support P2P energy trading can help meet local power needs while providing additional services like voltage control and reactive power support, which help reduce losses in transmission and distribution. Blockchain technology offers a secure platform for trading electricity as well as these extra services, ensuring safe and efficient transactions [22]

### **1.5 Literature Review**

In this section, a comprehensive study of the literature has been done to understand the background of related work. Related work has been categorized into different

categories based on the work presented in this thesis.

### 1.5.1 Blockchain and Smart Grid

This section outlines various works related to the integration of blockchain and SG infrastructure while also addressing fundamental challenges associated with this integration. The authors in [23, 24] discuss blockchain, which can enhance the resilience of SG by facilitating the secure exchange of distributed energy resources. In [25], the author explores blockchain, which can revolutionize the energy market, while [26] proposes blockchain as a tool for managing SG transactions.

The work in [27] presents a promising approach to integrating blockchain and IoT in SG, though it overlooks interoperability challenges between heterogeneous Internet of Things (IoT) devices and blockchain networks, which can hinder seamless data exchange.

A detailed taxonomy of blockchain, along with a review of consensus algorithms, applications, and technical challenges, is provided in [28]. Similarly, [29] presents a generalized blockchain architecture along with several consensus algorithms for deployment.

In [30], the author highlights the role of cryptocurrency in the energy sector. The importance of decentralization in SG is discussed in [31], advocating policies to support distributed energy resource integration.

Current blockchain-based energy projects are reviewed in [32], while [1] introduces NIST's Framework 3.0 to support blockchain interoperability. Trust in IoT networks is a key concern in SG environments, as addressed in [33] by presenting a layered architecture to enhance end-to-end trust within blockchain networks. In [34], a multi-dimensional review is provided on the technological advancements of blockchain in

SG, including use cases and real-world company projects.

The work in [35] investigates the integration of blockchain across SG applications, analyzing its architecture, potential benefits, and key challenges, including scalability, regulatory uncertainty, and interoperability. Similarly, [36] presents an in-depth analysis of blockchain's potential in SG but fails to fully address scalability and regulatory hurdles. Author [37] explores blockchain's role in SG but lacks a critical examination of the security vulnerabilities inherent in such systems. Likewise, [38] provides an overview of blockchain and SG but overlooks regulatory and governance issues that limit real-world applicability.

In [39] author proposes a permissioned blockchain with deep learning for the health-care system. The authors in [40] highlight blockchain's role in enhancing grid modernization and cybersecurity but do not thoroughly address its energy consumption and scalability limitations. In [41], the authors focus on improving the resilience of SG via blockchain but fall short in addressing privacy and data protection concerns, which are critical to public acceptance. A unique perspective is provided in [42], where the intersection of blockchain with both SG and healthcare sectors is explored. However, the paper lacks in-depth security analysis specific to healthcare data, raising trust issues in sensitive environments.

In [43], the Smart Grid–Cyber Physical System (SG-CPS) framework is introduced, featuring a standardized protocol with support for validation, testing, and design. The efficacy of SG-CPS was evaluated using real-world datasets such as the car-hacking dataset and the UNSW-NB15 dataset. These datasets, which include various types of attacks like spoofing, flooding, and replay attacks along with benign traffic, were used to validate the proposed security framework. Wireless Sensor Networks (WSNs), which are often integrated into smart grids, present several challenges. As

noted in [44], (SNs) can act as potential attack entry points. While periodic power cycling may extend the lifetime of SNs, it introduces latency and routing overhead. Additionally, the cost-effectiveness of deploying WSNs remains a critical concern due to the use of low-energy, short-range devices.

Finally, in [45], the authors introduce the Message Queuing Telemetry Transport (MQTT) protocol, a lightweight application-layer protocol popular in IoT systems. As IoT adoption increases, MQTT has become a target for security threats. In addition to discussing MQTT vulnerabilities, [46, 47] stresses the importance of strong cybersecurity measures in IoT-enabled SG. The above research work lacks unified, scalable, and intelligent access control for secure data exchange across decentralized grids.

### 1.5.2 Intrusion Detection System

After reviewing papers related to blockchain integration with smart grids, the research focused on various Intrusion Detection Systems (IDSs) that have been implemented in smart grid infrastructure to enhance security. The concept of intrusion detection has been widely implemented since it was first put forward by Anderson[48]. Most initial intrusion detection methods in the field of AMI intrusion detection used statistical techniques [49]. However, with the development of artificial intelligence technology, increasingly more Machine Learning (ML) and Deep Learning (DL) methods have been applied. With the development of machine learning and deep learning, many classifiers, such as Support Vector Machines (SVMs) [50, 51], decision trees [52], Extreme Gradient Boosting (XGBoost) [53–56], and Long Short-Term Memory (LSTM) [57–59], have been adopted in intrusion detection for SG. The author in[60] proposes a forward detection model of electricity theft based on

consumption patterns. A real-time Distributed Intrusion Detection System (DIDS) suitable for AMI was proposed in [61]. This model uses unsupervised online small-batch k-means clustering technology to monitor the data flow in AMI. The combined DT-SVM classifier is capable of accurately detecting real-time power theft at all levels of power transmission and distribution proposed in [62]

The research work [63] presents a systematic and automated approach to building a hybrid IDS that learns temporal state-based specifications for power scenarios, including disturbances, normal control operations, and cyberattacks. The work in [64] proposes a sequential pattern mining approach to accurately extract patterns of power-system disturbances and cyberattacks from heterogeneous time-synchronized data, including synchrophasor measurements, relay logs, and network event monitor logs. For instance, [65] introduced a Deep Blockchain Framework (DBF) aimed at enabling decentralized intrusion detection and safeguarding privacy in IoT networks by leveraging smart contracts.

To address the limitations of centralized IDS, especially for resource-constrained devices, the authors in [66] proposed two models: semi-distributed and fully distributed IDS. These approaches leverage efficient feature extraction and selection techniques and exploit the potential of fog-edge coordinated analytics. In another work, three advanced deep learning algorithms were applied for intrusion classification: Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and a hybrid CNN-LSTM model [67]. Furthermore, the author [68] introduced a fog-based, privacy-preserving, distributed, signature-based IDS, providing enhanced protection for edge devices.

The author [69–71], developed a machine learning-driven Intrusion Detection System (IDS) for enhancing SG security, combining Principal Component Analysis (PCA),

Linear Discriminant Analysis (LDA), and Support Vector Machine (SVM) to create a sophisticated intrusion detection mechanism. Additionally, a new optimization-based IDS was presented in [72], offering improved detection efficiency.

The challenges in implementing IDS for SG were discussed in detail in [73, 74], highlighting practical deployment issues. A systematic and deep analysis of rule-learning techniques and their applicability to IDS in SG was presented in the work, concluding with the identification of key criteria for effective rule generation and assessment. Moreover, [75] conducted an in-depth analysis of cyberattacks targeting IEC-61850 substations and reviewed recent efforts in detecting and preventing such threats. A major contribution of this work is the proposal of a novel taxonomy addressing both design and evaluation aspects of substation-specific IDSs.

While [76] provides traditional network security measures such as blocklists and firewalls that are still in use. Artificial Intelligence (AI)-driven techniques offer a dynamic alternative, capable of adapting to evolving threats without relying on predefined rules. The effectiveness of these AI-based systems significantly depends on the optimal selection of hyperparameters, which guide the learning behavior of the models. The authors in [77] presented an (IDS) using ensemble learning (i.e., bagging-, boosting-, and stacking-based) that can identify unknown threats, vulnerabilities, and cyberattacks in the SG environment. An IDS is a comprehensive solution that identifies harmful activity and complex cyberattacks, including zero-day assaults that are undetectable by other means [77]. The author in [78] designed a feature-based IDS that used particle swarm optimization for feature selection and random forest and neural networks for intrusion detection. However, the proposed approach was evaluated using obsolete datasets, called KDD99 and NSL-KDD. These datasets lack modern attacks that are mostly found in the SG environment. The author in

[79] designed an IDS based on extreme gradient boosting (XGBoost) and (LSTM). Although the proposed scheme obtained high values for accuracy and detection rate, it was evaluated using the NSL-KDD dataset [80–83]. The author in [84, 85] used BoT IoT dataset for the testing of the algorithm. The author in [86], used the conditional Deep Belief Network (DBN) to identify false injection attacks in SG. This detection model requires few external conditions and can achieve high accuracy. A wide and deep CNN model for the detection of electricity theft was proposed in [87]. The model consists of a deep component and a wide component, which are, respectively, used to identify the periodicity of electricity consumption and capture the global characteristics of electricity consumption data. The author in [88] proposed a Hybrid Deep Neural Network (HDNN) intrusion detection model by combining the CNN, the Gated Recursive Unit (GRU), and the Particle Swarm Optimization (PSO) algorithm. The model uses the CNN for feature selection and extraction, and GRU-PSO technology is used to classify the provided data. The system can automatically perform the processes of feature extraction and classification. In the research work of [89], a CNN was used to identify an intrusion, and the accuracy of the model was improved via data enhancement technology. In [90] the author adopted an autoencoder (AE) to reduce the dimension of the data to decrease the interference of redundant features, and a CNN was adopted to identify the intrusion information. In the research work of [91], an improved CNN was adopted to identify intrusion information. The CNN is improved to extract features across layers, and feature fusion is used to obtain comprehensive features. The author in [92] applied an Extreme Learning Machine (ELM) to intrusion detection, which was found to improve the detection speed and generalization ability of the model. In the research work of [93], an SG intrusion detection model that combines the Genetic Algorithm

(GA) and ELM was proposed. The model retains the advantages of the ELM, and the GA is introduced to ensure the optimal parameters of the model. The research work of [94, 95] introduced LSTM into the field of intrusion detection, explored the correlation of the temporal domain of intrusion information, and effectively reduced the rate of false positives. A Bidirectional GRU (BiGRU) was used in the research of [96] to detect abnormal data. Thus, the work presented in the above literatures lacks collaborative IDS and distributed intelligence sharing for the SG network

### 1.5.3 Privacy Preserving Scheme

In light of the evolving threats facing the SG, there is a pressing need for timely, accurate, and consistent security solutions, as highlighted in the work of [97]. To address these concerns, various security mechanisms have been developed, focusing on Authentication and Key Agreement (AKA) protocols and blockchain-enabled networks.

The AKA protocol is a prominent security method that uses symmetric cryptography and one-time passwords based on a challenge-response system. It ensures confidentiality of user identities and supports features like credential flexibility, resistance to Denial of Service (DoS) attacks, mutual authentication between devices, and secure session key exchange.

Numerous AKA-based security protocols have been proposed in literature. For example, the work of [98] proposed a security model, called “PBEM-SGN,” by combining permissioned blockchain with edge computing. This scheme used group signature-based authorization techniques and smart contracts running on the blockchain to provide energy security and data privacy protection. The work of [99] proposed a blockchain-based security scheme. ECCAuth [100] utilizes Elliptic Curve Cryptog-

raphy (ECC) to establish secure communication between SM and Service Providers (SP) after mutual authentication. AAS-IoTSG [101] combines ECC with Schnorr's signature for efficient key agreement in SG networks. TUAS-RESG [102] a three-factor user authentication scheme that integrates ECC, bitwise XOR operations, and one-way hash functions. Trusted Authority-Based Scheme (TABS) [103] designed for enabling secure authentication between SM and SP with centralized oversight. However, AKA-based systems face significant limitations, including centralized control, lack of transparency, difficulties in auditing and transaction verification, and poor interoperability [104].

Blockchain integrates to mitigate the challenges; decentralized security architectures, notably those based on blockchain technology, have gained attention. [105].

Recent blockchain-based solutions include the BPPS Scheme [106, 107] a privacy-preserving authentication scheme where SPs assign unique IDs to SMs and Demand Response Control Units (DRCUs) prior to deployment. Blockchain with Edge Computing [108] integrates permissioned blockchain with edge computing to address privacy and energy security challenges.

While these systems improve upon traditional AKA methods, many lack robust data vulnerability analysis, potentially compromising the integrity of ledger-stored information.

EFECT Scheme [109] enables efficient and privacy-preserving data aggregation with authentication. Survey on Privacy-Preserving Techniques [110] explores techniques in data aggregation, secure data trading, and privacy-aware analytics. Cross-Layer Security Analysis [111] addresses IoT integration and holistic security issues. Independent Component Analysis (ICA) [112] applies ICA for transforming raw data into privacy-preserving formats while retaining utility.

Bubbles of Trust [113] proposes a decentralized identification and authentication system. Blockchain and IoT Framework [114] offer secure data sharing, decentralized payment, and scalability. PPFSCADA [115] a privacy-preserving SCADA framework using permutation and clustering to anonymize traffic data. SPB Routing [116] a blockchain routing method using destination public keys to replace Trusted Third Parties (TTPs). Homomorphic Encryption and Deep Learning [117] a privacy-preserving aggregation model designed for high-load prediction tasks.

Decentralized Energy Trading [118] a blockchain solution using multi-signatures and anonymous messaging for secure peer-to-peer energy negotiation. The Lightweight Aggregation Scheme [119] combines homomorphic encryption with Password-Authenticated Key Exchange (PAKE). Bi-directional Communication Privacy [120] explores privacy implications in two-way SG communications. Power Injection Query Scheme [121] leverages AMI and LTE for secure querying of power injection data. The CBSRE Scheme [122] proposes a lightweight certificate-based signcryption with proxy re-encryption for SG-based IoT, using hyperelliptic curves. BPRF Framework [123] a blockchain-based reputation management framework for anonymous identity tracking and verification. Blockchain-Based Data Integrity Service [124] offers integrity assurance for both data owners and consumers without TTP. EBDA Framework [125] a lightweight, edge blockchain-assisted data aggregation architecture for SG.

Privy Sharing Framework [126] a blockchain-based system for privacy-preserving IoT data sharing in smart cities. The author in [127] proposes public key security protocols. Further in, [128] proposes autoencoder IDS.

#### 1.5.4 Smart Metering Infrastructure

In this section, we discuss various research efforts related to Advanced Metering Infrastructure (AMI) and its security enhancements in the SG environment. In [129], author proposes a cloud-centric collaborative security service placement for advanced metering infrastructure. The author in [130] introduced the concept of a trusted smart meter, utilizing remote anonymous attestation based on trusted computing to protect the private information of consumers at the terminal end. Building upon this, the author in [131] proposed an efficient, privacy-aware, multi-factor authenticated key establishment scheme (PMAKE). A notable feature of PMAKE is its ability to ensure the physical security of SM.

Further research [132] introduced a privacy-preserving communication framework for AMI with the following properties: A protracted publisher key is assigned to each smart meter by the Head End System. The encryption key for every message sent from the smart meter is derived from the publisher key. Each message includes a Message Authentication Code (MAC) to authenticate the smart meter identity. According to [106], a typical SG is composed of four main components: sensing, communication, control, and actuation systems. The SMs used by consumers—primarily include sensing and communication modules. In contrast, service providers (prosumers) operate with communication, control, and actuation modules. Additionally, the Demand Response Control System (DRCS) is responsible for evaluating the electric capabilities of SG networks and delivering effective Demand Response (DR) services and real-time pricing, as discussed in [103, 133].

Despite the various benefits offered by SG to utilities, service providers, and energy consumers, one of the primary challenges is the secure and reliable operation of the grid. Particularly, secure data sharing between SM and backend service servers re-

mains a significant concern. Even when power line communications are used (which are typically wired), vulnerabilities persist due to the use of open and unsecured wireless communication channels between SM and control systems [134]. In the absence of robust security measures, these communication channels are vulnerable to various cyber threats, including: replay attacks, impersonation, man-in-the-middle attacks, false data injection, and physical capture of SM. These threats, and the need for mitigating them, have been thoroughly analyzed in [135].

## 1.6 Research Gap

Some of the gaps identified in the literature are as follows:

1. Absence of unified, scalable, and intelligent access control for secure data exchange across decentralized grids.
2. Insufficient privacy-preserving mechanisms balancing performance and anonymity in Vehicle to Grid (V2G) communications.
3. Inefficient storage strategies for large-scale grid data, lacking hybrid on-off-chain verification mechanisms.
4. Lack of collaborative IDS and distributed intelligence sharing for adaptive anomaly detection in AMI.

## 1.7 Motivation of the Research Work

The rapid modernization of power systems has transformed traditional electricity networks into highly interconnected CP-SG supported by millions of IoT sensors, SM, EV, and cloud-edge computing platforms. While this digital transformation enables real-time monitoring, intelligent control, and large-scale renewable integration,

it also significantly increases the system's exposure to cyber threats. Recent cyberattacks on power utilities across the world—including large-scale outages, ransomware incidents, false data injection attacks, and advanced persistent threats—demonstrate that the smart grid is now a prime target for adversaries. These attacks have shown the potential to compromise grid stability, manipulate operational data, endanger consumer privacy, and cause widespread socio-economic damage [136–138].

The motivation for this research arises from the growing gap between SG digitalization and its cybersecurity readiness. Existing security mechanisms such as traditional authentication frameworks, centralized key-management systems, and isolated intrusion detection solutions struggle to address the complexity and scale of modern cyber-physical operations. Current systems lack decentralized trust, comprehensive privacy protection, collaborative intelligence sharing, and the ability to process massive heterogeneous data streams in real time. The evolving nature of cyberattacks, including coordinated IoT botnets, advanced spoofing, meter manipulation, EV-based intrusions, and AMI exploitation, demands intelligent, adaptive, and scalable security architectures.

Therefore, there is a critical need for an integrated, multi-layered security framework that brings together deep learning-based anomaly detection, blockchain-enabled decentralized trust management, privacy-preserving data sharing, and collaborative intrusion detection across distributed grid components. Such a unified approach can provide timely threat detection, tamper-evident auditability, data confidentiality, and trustworthy communication essential for reliable SG operation.

This research is motivated by the necessity to address these unmet challenges through the design of a comprehensive cybersecurity architecture that ensures the resilience, scalability, and privacy of next-generation SG infrastructures.

## 1.8 Research Objective

1. Designing a secure data sharing framework in SG infrastructure
2. Developing a secure and privacy-preserving framework for IoT-enabled EVs
3. Designing a collaborative IDS and distributed off-chain and on-chain storage system in AMI.

## 1.9 Organization of the Thesis

Subsequent sections of this thesis are outlined as follows:

1. Chapter 2 presents a deep learning-based secure data sharing framework for smart grid environments, focusing on its architecture and effectiveness.
2. Chapter 3 introduces a secure and privacy-preserving framework tailored for IoT-enabled smart grid systems, addressing key concerns around data confidentiality and integrity.
3. Chapter 4 explores the role of metaverse technologies in SG, emphasizing their potential to enhance security and promote sustainable energy management.
4. Chapter 5 is the conclusion and future work.

## 1.10 Conclusion

This chapter established the foundational context for the research by highlighting the critical importance of cybersecurity in modern smart grid infrastructure. Three key research gaps—(1) the lack of unified secure data sharing mechanisms across decentralized smart grid networks, (2) insufficient privacy-preserving frameworks for

IoT-enabled electric vehicles, and (3) limited collaborative intrusion detection capabilities in Advanced Metering Infrastructure (AMI) were identified. The research objectives were clearly defined to address these gaps through the design of blockchain-enabled deep learning frameworks. The chapter outlined the thesis organization, demonstrating how each subsequent chapter will progressively address specific security challenges in SG systems, ultimately contributing to a comprehensive security ecosystem that integrates advanced AI and distributed ledger technologies.

## Chapter 2

# Deep-Learning Empowered Secure Data Sharing for Smart Grid Infrastructure

### 2.1 Introduction

SG infrastructures rely on large-scale, real-time data exchange among smart meters, substations, control centers, and distributed energy resources. The integrity, confidentiality, and trustworthiness of this data are critical for reliable grid operation. However, conventional centralized architectures are highly vulnerable to cyberattacks, insider manipulation, and single-point failures.

To address these challenges, this chapter proposes a Blockchain and Deep-Learning-Enabled Secure Data Sharing (DBSDS) Framework for SG environments.

The framework integrates:

Blockchain for decentralized trust, tamper-proof data storage, and auditability.

Deep learning-based intrusion detection for real-time cyberattack detection.

Cryptographic authentication and session key establishment for secure communication.

The primary objective of this chapter is to establish a trusted, attack-resilient, and scalable data-sharing architecture for the SG, which later serves as the core

security backbone for the EV and metaverse extensions in Chapters 3 and 4. .

## 2.2 System Models

The suggested DBSDS framework was created using two submodels (network and threat models). Below is an explanation of each one.

### 2.2.1 Network Model

The proposed framework has three main entities: SP, SM, and DRCU, as shown in Figure.2.1. Initially, SP registers and then pre-deploys SM and DRCU in the SG network to provide efficient electricity distribution services. Once the entities are registered, the communication starts.

In SG infrastructure, the SM are responsible for gathering various information, including electricity consumption, voltage levels, consumption behavior, and electricity usage time[129]. The collected data are sent to DRCU, which evaluates the electric capability of SG networks and provides efficient DR and real-time pricing. However, SM is vulnerable to various potential attacks, as it sends the above-mentioned sensitive data to DRCU via an open wireless network. A malicious attacker can intercept, modify, and replay such data to violate user privacy and can manipulate the transmitted data. To guarantee data integrity, a secure mutual AKA protocol between SM and DRCU is suggested. The data may then be transferred between the two entities using the session key after authentication has been achieved. The majority of DRCUs are run by outside suppliers, though, and a malicious DRCU may change or remove the real data. To store the data sent between SM and DRCU, blockchain technology is deployed.

In the suggested architecture, only authorized transactions can be performed and

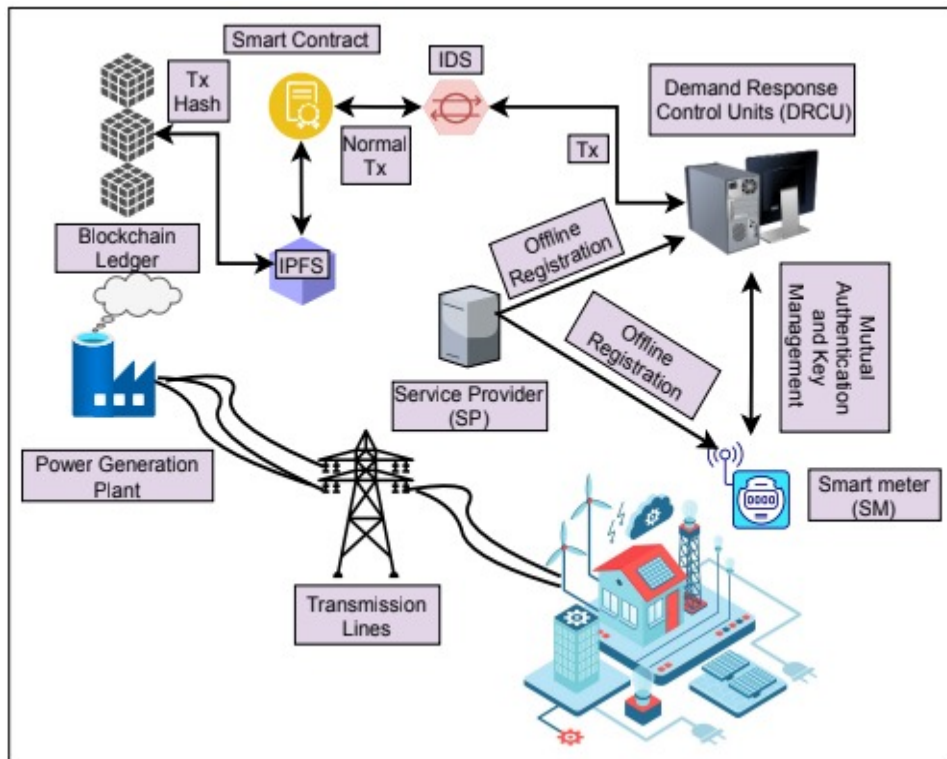


Figure 2.1: Network Model of Proposed Framework

be a part of the network, and a DL-based IDS for verification of the legal transactions sent by the SM is utilized. The proposed DL technique filters out anomalous transactions and delivers regular transactions to a Proof-of-Authentication (PoA) consensus process based on smart contracts.

The transaction is created, verified, and then appended over the Interplanetary file systems (IPFS) secure storage layer. The one-way cryptographic hash that is returned from the IPFS gets recorded over the blockchain ledger [105, 139]. As a result, the blockchain becomes lighter thanks to on-chain and off-chain techniques, which also cause block generation and access time to steadily decrease for models that keep entire data over the blockchain ledger[140].

### 2.2.2 Threat Model

The well-known “Dolev-Yao threat model (DY model)” was used to construct our suggested DBSDS framework [127]. In the well-known DY model, players share information across an unsafe channel; SMs and DRCUs are viewed as unreliable parties. A hacker has the power to eavesdrop, alter, or delete the sent communications because the conversation is taking place through an unsafe channel.

The entire SG infrastructure must be entirely trustworthy for the SP to exist. Additionally, because some SMs are physically captureable and cannot be continuously monitored, power analysis attacks can access the credentials stored in these physically captured SMs.

The “CK-adversary model: The attacker in the CK-adversary model has the same access to information as in the DY model, but he or she may additionally steal confidential information, which includes “private keys,” “session keys,” and “session states.” In addition to the qualities mentioned above, the system needs to be resistant to various cryptographic assaults.

## 2.3 The Proposed Framework

This section covers the core Intrusion Detection and Classification (RENS) architecture components used in designing the proposed DBSDS framework

### 2.3.1 RENS Architecture

The DL technique has the ability to adapt and work efficiently with high data volume and dimension. The proposed DL-based RENS has two core components:

First, a feature extraction model based on Variational AutoEncoder (VAE).

Second, the extracted features are used by the Attention-based Bidirectional Long

Short-Term Memory (ABiLSTM) technique to detect intrusions. Before discussing the above method, the concept of AE is revised.

An AE is a kind of neural network that learns without supervision. It takes input data and compresses it into a smaller, lower-dimensional form (encoding) before rebuilding the original data from this smaller form (decoding). An AE consists of three main parts: the encoder, the bottleneck (latent space), and the decoder. The encoder reduces the input data's dimensionality, capturing essential features. Different types of AE are

- a) Denoising Autoencoder
- b) Sparse Autoencoder
- c) Variational Autoencoder (VAE)
- d) Convolutional Autoencoder

Among the above, VAE has been used in the design of RENS because of the following features[128, 141].

- 1) VAEs are highly efficient for generative and feature-learning tasks since they represent input data as probability distributions rather than fixed points.
- 2) This probabilistic approach helps in capturing meaningful structures in the data, producing realistic and diverse outputs.
- 3) It is widely used in image generation, anomaly detection, and data synthesis due to its ability to generalize well and reduce overfitting.

### **2.3.2 RENS Variational AutoEncoder (VAE) for Feature Extraction**

The VAE network, in contrast to the autoencoder's low-dimensional embedding, offers a probabilistic aspect in the latent variable space, which can better capture the prominent characteristics of multivariate SG data. The VAE's main goal is to

describe the distribution of the original data set  $(X) = \{(X)_i\}_{i=1}^n$  using the latent variable  $(Z)$ . The conditional distribution of the latent variable  $(Z)$  is assumed to be Gaussian. The theory shows that using a neural network, the hidden variable  $(Z)$  that satisfies the Gaussian distribution may create data that satisfies any distribution. The latent variable  $(Z)$  creates a dataset  $\widehat{(X)} = \{\widehat{(X)}_i\}_{i=1}^n$  that is extremely identical to the original data  $(X) = \{(X)_i\}_{i=1}^n$  by optimizing the created parameter. As a result, there is a need to maximize marginal likelihood  $(P)_\theta((X))$  [142, 143].

$$(P)_\theta(X) = \int (P)_\theta((X) | (Z)) (P)_\theta((Z)) d(Z) \quad (2.1)$$

Where  $(P)_\theta((X) | (Z))$  denotes the posterior probability of  $(X)$  given  $(Z)$ , as described by a neural network with parameters  $\theta$ , and  $(P)_\theta(\mathcal{Z})$  is the a priori probability of latent variables as modeled. networks with parameters  $\theta$ . Decoders are the name for these neural networks. The Evidence Lower Bound (ELBO) from Jensen's inequality is the lower bound of log-likelihood  $\log \mathcal{P}_\theta((X))$  [143].

$$\begin{aligned} \log (P)_\theta((X)) &= \log (E)_{(Q)_\phi((Z)|(X))} \left[ \frac{(P)_\theta((X) | (Z)) (P)_\theta((Z))}{(Q)_\phi((Z) | (X))} \right] \\ &\geq (E)_{(Q)_\phi((Z)|(X))} \left[ \log \frac{(P)_\theta((X) | (Z)) (P)_\theta((Z))}{(Q)_\phi((Z) | (X))} \right] \\ &\equiv (L)((X); \theta, \phi) \end{aligned} \quad (2.2)$$

where  $(E)_{(Q)_\phi((Z)|(X))}$  the mathematical expectation,  $(Q)_\phi((Z | (X)))$  is the posterior probability of  $(Z)$  given  $(X)$ , represented by a neural network with parameters  $\phi$ , and  $(Q)_\phi((Z | (X)))$  is generally the normal distribution  $(N)((Z) | \mu_\phi(\theta)((X)), \sigma_\phi^2((X)))$ , where  $\mu_\phi(\theta)((X))$  and  $\sigma_\phi^2((X))$  are neural networks with argument  $\phi$  and input  $(X)$ , respectively. Encoders are the name for these neural networks.  $(L)((X); \theta, \phi)$  can be viewed as the lower bound of  $\log (P)_\theta((X))$ .

Thus, the ELBO can also be defined as [144]:

$$\begin{aligned}
 (L)_{VAE} &= (L)((X); \theta, \phi) \\
 &= (E)_{(Q)_{\phi}((Z|(X)))} [\log (P)_{\theta} ((X) | (Z))] \\
 &\quad - (D)_{KL} ((Q)_{\phi} ((Z | (X))) \| (P)_{\theta} ((Z)))
 \end{aligned} \tag{2.3}$$

Specifically, the loss function  $(L)_{VAE}$  in eq 2.3 can be divided into two parts: 1) a logarithmic reconstruction likelihood term  $(E)_{(Q)_{\phi}((Z|(X)))} [\log (P)_{\theta} ((X) | (Z))]$  and 2) a Kullback–Leibler (KL) divergence  $-(D)_{KL} ((Q)_{\phi} ((Z | (X))) \| (P)_{\theta} ((Z)))$ . The former term is used to reduce the reconstruction loss of the square  $L_2$  norm between  $(X)$  and  $(\widehat{X})$ , while the latter term is used to characterize the posterior distribution of  $(Q)_{\phi}((Z | (X)))$  the prior distribution  $(P)_{\theta}((Z))$ .

The VAE can be trained in an unsupervised manner. Thus, the input intrusion data is converted to a latent representation that is considered to lose the individual singular characteristic. The obtained latent features are forwarded to an ABiLSTM-based classifier to detect intrusions.

### 2.3.3 RENS VAE-ABiLSTM for Attack Detection

BiLSTM is a Recurrent Neural Network (RNN) version that comprises two Long Short-Term Memory (LSTM) networks that move forward and backward while storing time-series data [145, 146].

BiLSTM is appropriate for handling intrusion detection-related tasks and has been effectively used to attack event classification since intrusion data can be conveniently represented as time series and depends on the data in preceding frames.

The LSTM-based Recurrent Neural Networks (RNNs) are typically composed of four key components:

an input gate  $(I)_{(T)}$  associated with weight matrix  $(W)_{(I)}$  and bias  $(B)_{(I)}$ ,  
 a forget gate  $(F)_{(T)}$  with corresponding weight matrix  $(W)_{(F)}$  and bias  $(B)_{(F)}$ ,  
 and an output gate  $(O)_{(T)}$  governed by a weight matrix  $(W)_{(O)}$  and bias  $(B)_{(O)}$ .

These gates determine whether to accept new inputs, discard previously stored information, or output the updated state. The decisions made by these gates depend on the current input  $(X)_{(T)}$ , the hidden state from the previous time step  $(H)_{(T)-1}$ , and the previous cell state  $(C)_{(T)-1}$ . This behavior is mathematically expressed through the following set of equations:

$$(I)_{(T)} = \sigma \left( (W)_{(I)}(X)_{(T)} + (W)_{(I)}(H)_{(T)-1} + (W)_{(I)}(C)_{(T)-1} + (B)_{(I)} \right) \quad (2.4)$$

$$(F)_{(T)} = \sigma \left( (W)_{(F)}(X)_{(T)} + (W)_{(F)}(H)_{(T)-1} + (W)_{(F)}(C)_{(T)-1} + (B)_{(F)} \right) \quad (2.5)$$

$$(G)_{(T)} = \tanh \left( (W)_{(G)}(X)_{(T)} + (W)_{(G)}(H)_{(T)-1} + (W)_{(G)}(C)_{(T)-1} + (B)_{(G)} \right) \quad (2.6)$$

$$(C)_{(T)} = (I)_{(T)}(G)_{(T)} + (F)_{(T)}(C)_{(T)-1} \quad (2.7)$$

$$(O)_{(T)} = \sigma \left( (W)_{(O)}(X)_{(T)} + (W)_{(O)}(H)_{(T)-1} + (W)_{(O)}(C)_{(T)-1} + (B)_{(O)} \right) \quad (2.8)$$

$$(H)_{(T)} = (O)_{(T)} \tanh \left( (C)_{(T)} \right) \quad (2.9)$$

As a result, the weighted total will be calculated utilizing both the prior cell state and the current information supplied by the cell to generate the current cell state  $C_{\mathcal{J}}$ . It's useful to have access to both future and past context for many sequence modeling activities.

Unlike standard LSTM networks, which process input sequences strictly in chronological order and thereby overlook future context, Bidirectional LSTM (BiLSTM) networks address this limitation by incorporating an additional LSTM layer that operates in the reverse temporal direction. This design enables the model to cap-

ture dependencies from both past and future time steps by combining information processed in the forward and backward directions.

$$\overrightarrow{(H)}_{(T)} = (LSTM) \left( (X)_{(T)}, \overrightarrow{(H)}_{(T)} - \vec{1} \right) \quad (2.10)$$

$$\overleftarrow{(H)}_{(T)} = (LSTM) \left( (X)_{(T)}, \overleftarrow{(H)}_{(T)} - \vec{1} \right). \quad (2.11)$$

Finally, by concatenating the forward and backward states of BiLSTM, the result can be summarized as follows:

$$(H)_{(T)} = \left[ \overrightarrow{(H)}_{(T)}, \overleftarrow{(H)}_{(T)} \right] \quad (2.12)$$

The attention mechanism is added to the traditional BiLSTM model. The attention layer, in particular, adds larger weights to the input sequence's most relevant and essential features.

This technique has the advantage of preserving lengthier input sequences.

Let  $(H)$  denote a matrix containing the BiLSTM layer's output vectors  $[(H)_1, (H)_2, \dots, (H)_{(T)}]$ , where  $(T)$  is the feature length. A weighted sum of these output vectors is used to construct the input representation  $(R)$ :

$$(M) = \tanh((H)) \quad (2.13)$$

$$\alpha = (softmax) \left( (W)^{(T)}(M) \right) \quad (2.14)$$

$$(R) = (H)\alpha^T \quad (2.15)$$

where  $(W)^{(T)}$  denotes the transpose of the trained parameter vector. The final feature-pair representation form used for detecting and classifying attacks can be written as,

$$(H)^* = \tanh((R)) \quad (2.16)$$

To detect attacks, the last layer with the sigmoid activation function  $\mathcal{S}$  detects

attacks 0 and normal 1 using the below formula

$$(S)((H)^*) = \frac{1}{1 + e^{(H)^*}}, (S) \rightarrow [0, 1] \quad (2.17)$$

## 2.4 Deep Learning Analysis

In SG infrastructure, the SM gathers various information, including electricity consumption, voltage levels, consumption behavior, and electricity used time. The collected data are sent to DRCU, which evaluates the electric capability of SG networks and provides efficient demand response and real-time pricing. So the information that has to be protected is electricity consumption, voltage levels, consumption behavior and electricity used time, and real-time pricing. For this protection, the RENS scheme was discussed in the previous section. To check the feasibility of RENS, two types of data sets are used, which are described in the next section.

### 2.4.1 Description of Datasets

The experiment uses two recent intrusion datasets to validate the *RENS* performance. The corpus is available for free on open-source websites [83, 84].

### 2.4.2 ToN-IoT Dataset:

The ToN-IoT dataset is used to test the proposed *RENS* detection capability. The data is gathered from IoT networks using parallel processing to acquire a mix of routine and cyberattack events. Table 2.1 illustrates the functional distribution of normal and attack cases. There are 43 features in this dataset, including 300,000 normal observations and 161,043 attack observations [83].

### 2.4.3 BoT-IoT Dataset:

The BoT-IoT dataset is also used to evaluate the performance of the proposed *RENS*. This dataset was developed at the UNSW Canberra Cyber Range Laboratory. It has about 72 million recordings that include both simulated and real-time settings [84]. Table 2.2 shows the distribution of normal and attack cases. There are 16 features in this dataset, including 477 normal observations and 3,668,045 attack observations.

### 2.4.4 Smart Grid Dataset

The ToN-IoT and power system datasets are connected because they both describe cyber-physical behavior in modern smart grid environments. TON-IoT collects multi-modal information, such as IoT sensor telemetry, network traffic, and operating system logs, during both normal and cyberattack settings. Similarly, power system datasets include time-series readings from smart meters, PMUs, SCADA sensors, and substation equipment, which are increasingly linked via IoT-based communication networks. By mapping ToN-IoT device activity to corresponding smart grid components, investigations can be made into how cyber breaches in IoT networks affect the physical operation of power systems. This link enables the creation of unified intrusion detection and anomaly classification models capable of distinguishing between cyber-induced disruptions and true electrical problems, resulting in more robust cybersecurity and resilience measures for smart grids[147, 148].The realtion is illustrated in the Figure 2.2

### 2.4.5 Performance Analysis

The simulations were run on a computer with an NVIDIA GeForce GTX 1060 16 GB GPU, two Intel i5-3470 CPUs running at 2.20 GHz, 64 GB of RAM, and Windows 11

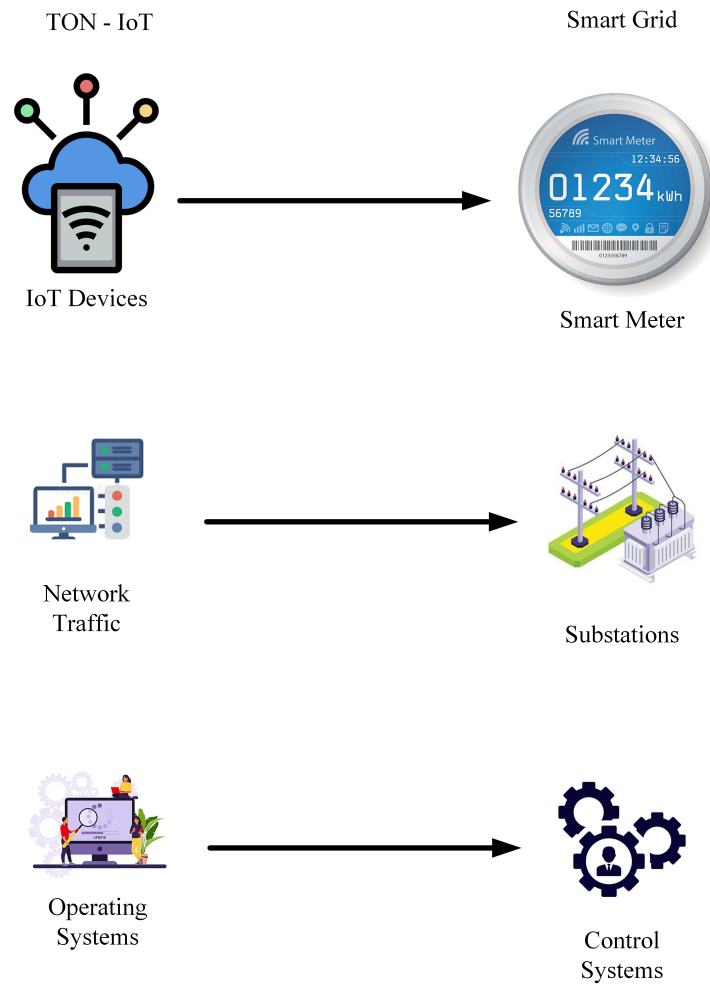


Figure 2.2: Relation between ToN-IoT data and Smart Grid Dataset

as the operating system. The textit-RENS method was created using the TensorFlow package Keras.

The smart contract module was developed on the Ethereum Rinkeby network[149]. While A-BiLSTM used two hidden layers with (16,8) neurons, the VAE layer was trained over 25 epochs using two hidden layers containing (128,64) neurons. The final layer has a sigmoid classifier, ReLU activation, Adam Optimizer, binary cross-entropy as the loss function, and a batch size of 100 over 10 epochs.

Table 2.1: Distributions of attacks and normal instances in ToN-IoT dataset

| Division of class | Total occurrences | Percentage of class frequency |
|-------------------|-------------------|-------------------------------|
| Training Set      |                   |                               |
| Normal            | 209801            | 65.069%                       |
| Attack            | 112929            | 34.931%                       |
| Testing Set       |                   |                               |
| Normal            | 90199             | 65.069%                       |
| Attack            | 48114             | 34.931%                       |
| Total             | 461043            | 100%                          |

Table 2.2: Distributions of attacks and normal instances in BoT-IoT dataset

| Division of class | Total occurrences | Percentage of class frequency |
|-------------------|-------------------|-------------------------------|
| Training Set      |                   |                               |
| Normal            | 370               | 65.069%                       |
| Attack            | 2934447           | 34.931%                       |
| Testing Set       |                   |                               |
| Normal            | 107               | 65.069%                       |
| Attack            | 733598            | 34.931%                       |
| Total             | 3668522           | 100%                          |

#### 2.4.6 Data Preprocessing

The intrusion datasets were preprocessed using various steps. First, the categorical values were converted into numeric data using label encoding. The numeric values were of different ranges; therefore, the min-max normalization technique was used to scale the values between 0 and 1. The ToN-IoT dataset was then divided into a 70-30 ratio of training and testing sets. On the other hand, for the BoT-IoT dataset, two separate files of training and testing were downloaded and were used to perform experiments [150, 151].

#### 2.4.7 Attack Detection Analysis

The main objective of this subsection is to assess the significance of the proposed IDS, i.e., *RENS*. Different evaluation metrics such as accuracy vs. loss for VAE technique, confusion matrix, ROC curve,  $\mathcal{AC}$ ,  $\mathcal{PR}$ ,  $\mathcal{DR}$ ,  $\mathcal{F1}$ , and  $\mathcal{FAR}$  score. In Fig. 2.3 and Fig. 2.4 the accuracy vs. loss is illustrated for the ToN-IoT as well as the BoT-IoT

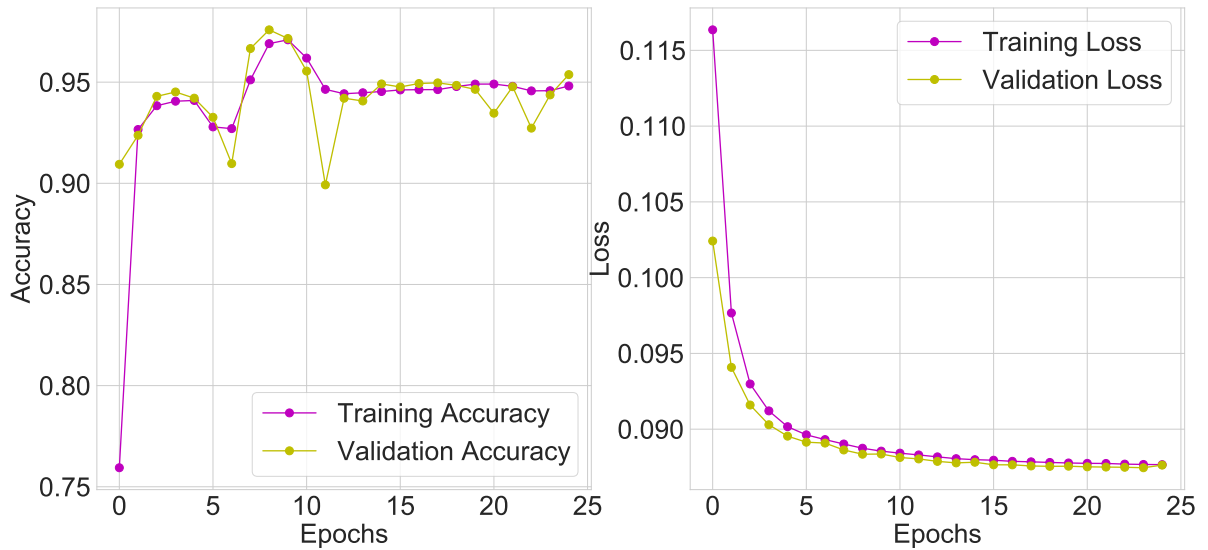


Figure 2.3: The accuracy vs. loss for VAE technique using ToN-IoT dataset

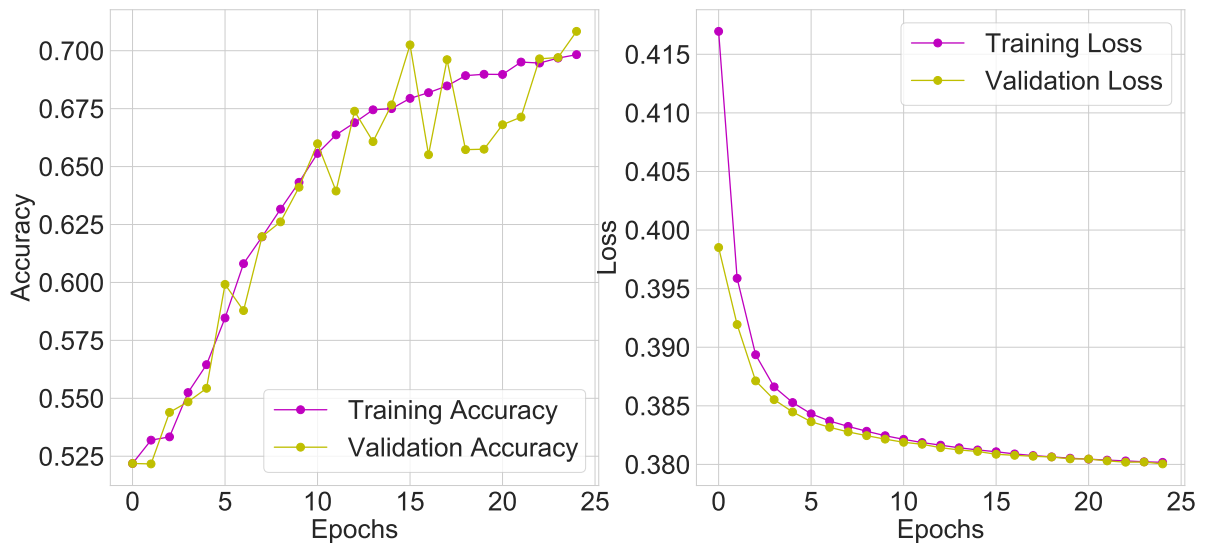


Figure 2.4: The accuracy vs. loss for VAE technique using BoT-IoT dataset

datasets, respectively. It is seen that the VAE-based feature extraction module has learned both datasets well and obtained *val\_loss* of 0.0876 and *val\_accuracy* of 95.37, and *val\_loss* of 0.3809 and *val\_accuracy* of 70.67 using the ToN-IoT as well as the BoT-IoT datasets, respectively. The extracted features from this module are used to detect intrusions.

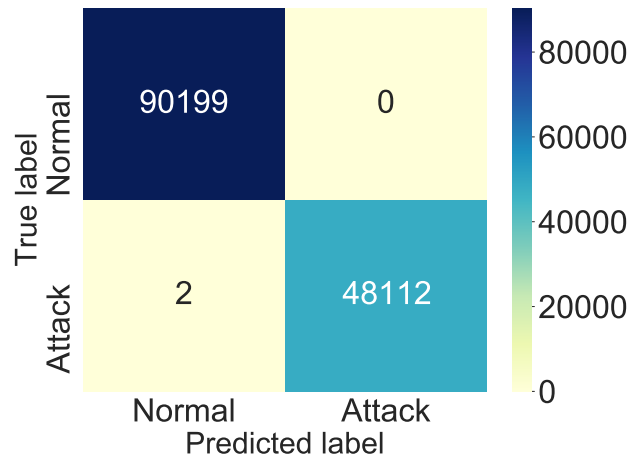


Figure 2.5: The confusion matrix obtained from RENS for ToN-IoT dataset

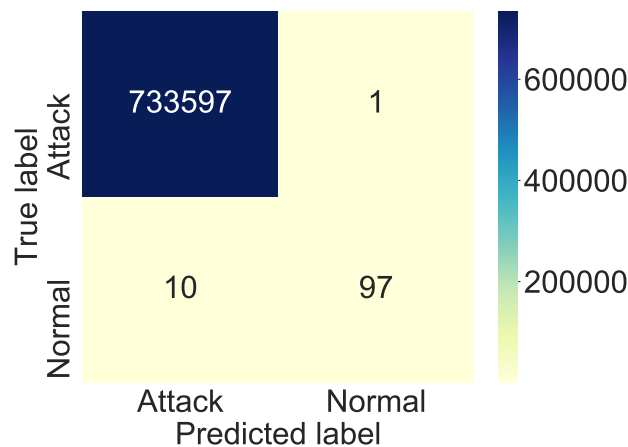


Figure 2.6: The confusion matrix obtained from RENs for BoT-IoT dataset

The confusion matrix (CM) is a valuable tool that contains information about an IDS model's efficiency in binary classifications. The  $\mathcal{TP}$ ,  $\mathcal{FN}$ ,  $\mathcal{FP}$ , and  $\mathcal{TN}$  are all represented in the matrix. It's worth noting that "positive" implies the IDS projected the traffic record as attack data, while "negative" means the IDS anticipated the traffic record as normal data. Figure. 2.5 and 2.6 show the CM obtained from our proposed *RENS*-based IDS. It is observed that for the ToN-IoT dataset in Figure.2.5 only 2 attack instances were misclassified. On the other hand, for the BoT-IoT dataset in Figure.2.6 1 attack and 10 normal instances were misclassified. The lesser

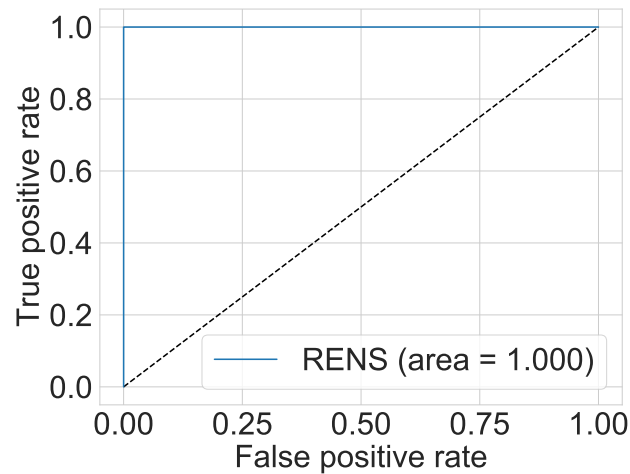


Figure 2.7: The AU-ROC Curve obtained for ToN-IoT dataset

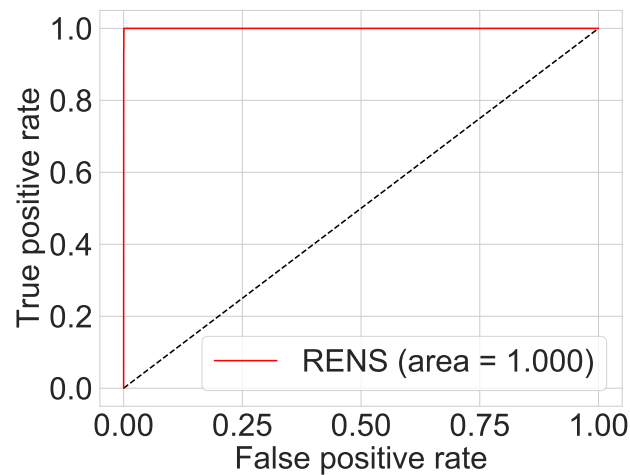


Figure 2.8: The AU-ROC Curve obtained for BoT-IoT dataset

the misclassification, the better the performance is considered.

A Receiver Operating Characteristic (ROC) curve is a two-dimensional depiction of a classifier's recall and fall-out scores. The amount of genuine negatives leaking in as  $\mathcal{F}\mathcal{P}s$  is referred to as "fall-out." It's the number of  $\mathcal{F}\mathcal{P}s$  divided by the number of genuine negatives. The number  $\mathcal{F}\mathcal{P}s$  obtained by the classifier is referred to as recall. While the classifier can accurately extract data, we don't want data from the negative lot to seep in. The classification threshold is a set of probabilities used to discriminate between two classes using a classifier's output. For example,

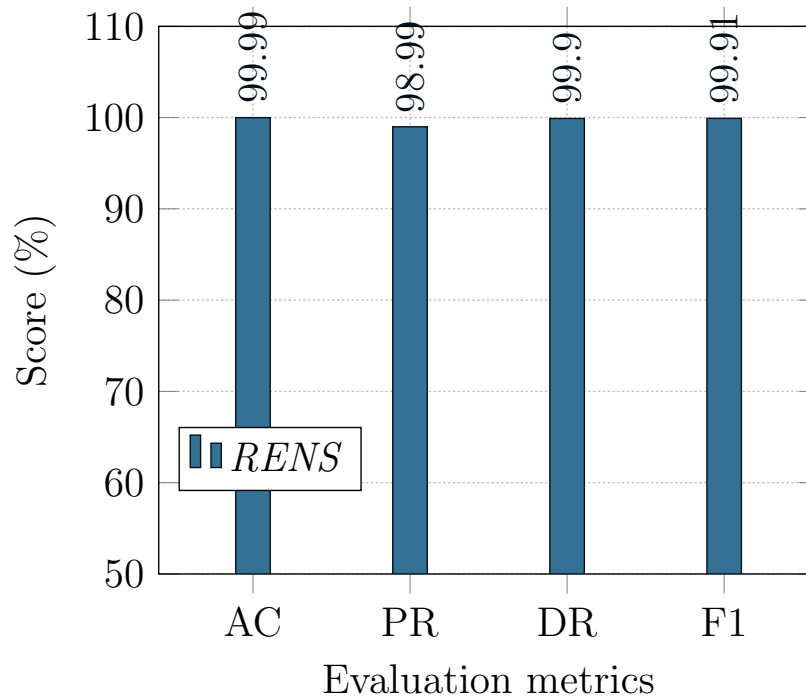


Figure 2.9: Performance of proposed *RENS* on different evaluation metrics using ToN-IoT dataset

if the threshold is 0.5 and the classifier produces 0.48, the prediction is regarded as negative, whereas 0.65 indicates positive. As a result, the trade-off between the two metrics for picking an acceptable threshold can be seen on the ROC curve at locations corresponding to particular classification thresholds and their related TPR and FPR values. Figure.2.7 and 2.8 depict the ROC curve obtained from the proposed *RENS*-based IDS. It is seen in Figure.2.7 and Figure.2.8 that for both datasets the *RENS* has obtained an AUC value of 1; this represents the classifiers' ability to differentiate between the two classes.

Finally, the results are obtained for both datasets in terms of  $\mathcal{AC}$ ,  $\mathcal{PR}$ ,  $\mathcal{DR}$  and  $\mathcal{F1}$  score. Figure. 2.9 shows these values using the ToN-IoT dataset, and Figure. 2.10 shows the BoT-IoT dataset. It is seen that the proposed *RENS* has achieved 99.99%  $\mathcal{AC}$ , 98.99%  $\mathcal{PR}$ , 99.90

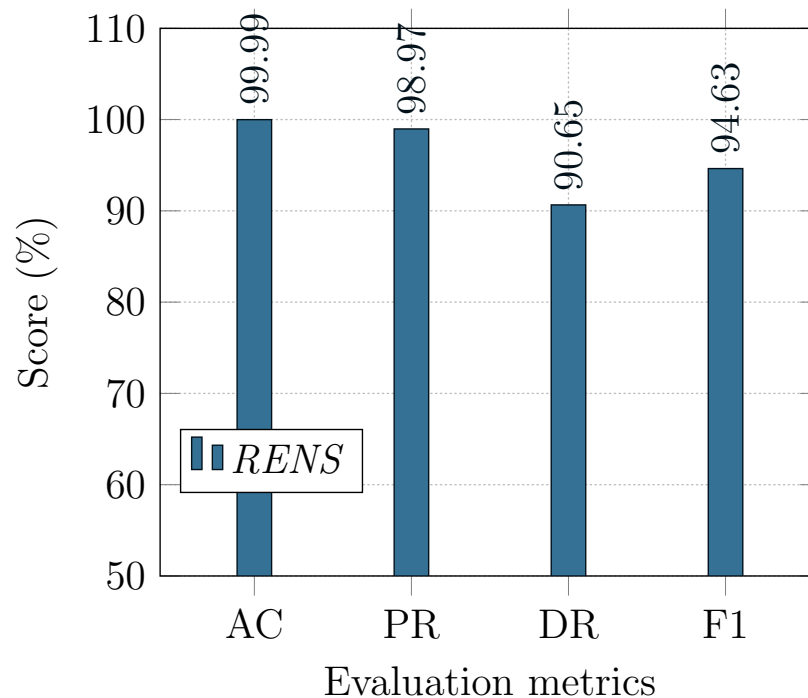


Figure 2.10: Performance of proposed *RENS* on different evaluation metrics using BoT-IoT dataset

#### 2.4.8 Comparison with existing Approaches

The performance of the proposed *RENS*-based IDS is evaluated and compared with some baseline algorithms on BoT-IoT and ToN-IoT datasets with respect to  $\mathcal{AC}$ ,  $\mathcal{PR}$ ,  $\mathcal{DR}$  and  $\mathcal{F1}$  score.

The descriptions of baseline algorithms are as follows: Naive Bayes (NB): Naive Bayes is a simple and effective supervised learning algorithm used for classification tasks in machine learning. It applies Bayes' Theorem with a strong, "naive" assumption that all features are independent of each other given the class label [152, 153].

Decision Tree (DT): A decision tree is a supervised learning algorithm used for both classification and regression tasks, structured as a tree of decision nodes and leaf nodes. It splits a dataset into subsets based on feature values, recursively building tree branches that represent decision paths.

Random Forest (RF): Random Forest is a powerful supervised machine learning algorithm that builds an ensemble (collection) of many decision trees to improve prediction accuracy and robustness.

Artificial Neural Network (ANN): An Artificial Neural Network (ANN) is a computational model inspired by the structure and function of the human brain, designed to process information, recognize patterns, and make predictions.

To ensure a comprehensive and scientifically valid evaluation of the proposed intrusion detection framework, the performance was compared against a diverse set of benchmark models representing all major categories of machine learning and deep learning algorithms used in smart grid and IoT security research. NB, DT, and RF were selected as classical machine-learning baselines because they are widely adopted in intrusion detection studies and provide strong reference points for probabilistic, rule-based, and ensemble learning performance, respectively. Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) networks were included to benchmark the system against modern deep learning architectures, with ANN serving as a nonlinear feature learner and LSTM capturing temporal dependencies commonly found in cyberattack patterns.

Table 2.3 shows the comparison using the ToN-IoT dataset. It can be seen that NB model achieved 90.69%  $\mathcal{AC}$ , 77.70%  $\mathcal{DR}$ , 77.68%  $\mathcal{PR}$  and 72.43%  $\mathcal{F1}$  scores. Similarly, DT achieved 95.34%  $\mathcal{AC}$ , 80.00%  $\mathcal{DR}$ , 74.42%  $\mathcal{PR}$  and 76.33%  $\mathcal{F1}$  scores. The RF model obtained 97.81%  $\mathcal{AC}$ , 85.43%  $\mathcal{DR}$ , 87.55%  $\mathcal{PR}$  and 86.41%  $\mathcal{F1}$  scores. Finally, the proposed *RENS* achieved the highest values for these parameters, i.e., 99.99%  $\mathcal{AC}$ , 98.99%  $\mathcal{DR}$ , 99.99%,  $\mathcal{PR}$  and 99.91%  $\mathcal{F1}$  scores.

Similarly, Table 2.4 shows the comparison based on the BoT-IoT dataset. It is seen that the NB model achieved 94.69%  $\mathcal{AC}$ , 80.70%  $\mathcal{DR}$ , 82.18%,  $\mathcal{PR}$  and 74.53%  $\mathcal{F1}$

Table 2.3: Performance comparison with baseline algorithms using ToN-IoT dataset

| Techniques         | $\mathcal{AC}$ | $\mathcal{DR}$ | $\mathcal{PR}$ | $\mathcal{F1}$ |
|--------------------|----------------|----------------|----------------|----------------|
| <b>NB</b>          | 90.69%         | 77.70%         | 77.68%         | 72.43%         |
| <b>DT</b>          | 95.34%         | 80.00%         | 74.42%         | 76.33%         |
| <b>RF</b>          | 97.81%         | 85.43%         | 87.55%         | 86.41%         |
| <b>ANN</b>         | 99%            | 99%            | 99%            | 99%            |
| <b>LSTM</b>        | 99%            | 99%            | 99%            | 99%            |
| <b><i>RENS</i></b> | 99.99%         | 98.99%         | 99.99%         | 99.91%         |

Terms & Abbreviations: AC: Accuracy; DR: Detection Rate; PRPR: Precisionate; FAR: False Alarm Rate; NB: Naive Bayes; DT: Decision Tree; RF: Random Forest;

Table 2.4: Performance comparison with baseline algorithms using BoT-IoT dataset

| Techniques         | $\mathcal{AC}$ | $\mathcal{DR}$ | $\mathcal{PR}$ | $\mathcal{F1}$ |
|--------------------|----------------|----------------|----------------|----------------|
| <b>NB</b>          | 94.69%         | 80.70%         | 82.18%         | 74.53%         |
| <b>DT</b>          | 94.30%         | 79.82%         | 81.42%         | 85.33%         |
| <b>RF</b>          | 95.45%         | 84.88%         | 85.85%         | 87.22%         |
| <b>ANN</b>         | 99%            | 99%            | 92%            | 96%            |
| <b>LSTM</b>        | 99%            | 99%            | 92%            | 94%            |
| <b><i>RENS</i></b> | 99.99%         | 98.97%         | 90.65%         | 94.63%         |

scores. Similarly, DT achieved scores of 94.30%  $\mathcal{AC}$ , 79.82%  $\mathcal{DR}$ , 81.42%,  $\mathcal{PR}$  and 85.33%  $\mathcal{F1}$  score. The RF model obtained 95.45%  $\mathcal{AC}$ , 84.88%  $\mathcal{DR}$ , 85.85%,  $\mathcal{PR}$  and 87.22%  $\mathcal{F1}$  scores. Finally, the proposed *RENS* achieved the highest values for these parameters, i.e., 99.99%  $\mathcal{AC}$ , 98.97%  $\mathcal{DR}$ , 90.65

The reasons for this better performance of the proposed *RENS* are as follows. First, the VAE technique extracted meaningful patterns in an unsupervised manner. These hidden and unknown patterns were then converted/transformed into a new compressed form.

The A-BiLSTM added additional weights to the most important and relevant features and processed the time-series datasets in both forward and backward directions. Furthermore, it also stored useful data and knowledge for a long period while discarding unnecessary data.

## 2.5 Blockchain Scheme

In this subsection, the mutual authentication, key agreement, and smart contract-based consensus mechanism for a secure, traceable, and transparent data storage process are discussed.

### 2.5.1 System Initialization Phase

This section describes the initial setup process for network entities. The configuration is performed by a trusted authority, denoted as  $SP$ , who is responsible for generating and distributing the necessary network parameters for secure communication. *Step-1:* The trusted authority  $SP$  selects an elliptic curve with a non-singular property defined as

$E_n(a, b)$  with the form  $y^2 = x^3 + ax + b \pmod{g}$  over the real Galois field  $GF(g)$ , where  $g$  denotes a large prime number. To ensure the curve is non-singular, it must satisfy the condition  $4a^3 + 27b^2 \neq 0 \pmod{g}$  [154]. This condition ensures the elliptic curve does not have any cusps of self-intersection and therefore maintains its cryptographic strength. The curve also satisfies the non-singularity property,  $\omega$  which includes both the identity (point at infinity) and the zero point.

Next, the  $SP$  selects the initial point  $BS \in E_n(a, b)$  with the closest order of  $g$ , say  $n$  i.e.,  $n \cdot BS = \omega$ , where  $n \cdot BS$  signifies the scalar multiplication point over the elliptic curve, and  $n \in Z_g$  is the discrete algorithm over the base point  $BS$ .

*Step-2:* The  $SP$  computes a cryptographic hash function with one-way working principles that is collision resistant,  $hfs(.)$ . This is computed by the secure hash algorithm (SHA-256) for the reason of security that provides a 256-bit message digest.

*Step-3:* The  $SP$  chooses the unique identity  $ID_{SP}$  and performs computation of the master key  $M_{SP}$  and produces a private key randomly  $PR_{SP} \in Z_g$ , where  $Z_g =$

$\{1, 2, 3, 4, \dots, g-1\}$ . The  $SP$  computes a public key i.e.;  $PB_{SP} = PR_{SP} \cdot BS$ .

*Step 4:* The  $SP$  preserves a  $PR_{SP}$  and  $M_{SP}$  preserves the secret key and shares the public parameter like  $\{E_n(a, b), BS, PB_{SP}, hfs(.)\}$ .

### 2.5.2 Registration Phase

This phase illustrates the registration phase of entities of a network.

(i) *DRCU Registration:* The  $SP$  registers the Demand Response Control Units ( $DRCU$ ) using the following steps, which are illustrated below.

*Step-1:* The  $SP$  selects a unique identity  $ID_{DRCU}$  and performs computation of pseudo identity  $SID_{DRCU} = h(ID_{SP} || M_{SP} || RT_{DRCU})$ , where  $RT_{DRCU}$  denotes the registration timestamp of Demand Response Control Units. Further,  $SP$  chooses a temporary identity  $TIDM_{DRCU}$  and computes a random secret value  $PR_{DRCU} \in Z_g$  and finds a public key  $PB_{DRCU} = PR_{DRCU} \cdot BS$ .

*Step-2:* The  $SP$  generates a certificate of  $DRCU$  as  $CRT_{DRCU} = M_{SP} + hfs(PB_{SP} || PB_{DRCU} || ) * PR_{SP} \pmod{g}$ . Further,  $SP$  stores the  $DRCU$  information (i.e.,  $(TIDM_{DRCU}, SID_{DRCU}, CRT_{DRCU}, PR_{DRCU}, E_n(a, b*), hfs(.))$ ) into the memory and distributes the public key  $PB_{CLS}$  for further access.

(ii) *SM Registration:* The  $SP$  registers the Demand Response Control Units ( $SM$ ) using the following steps, which are illustrated below.

*Step-1:* The  $SP$  picks a unique identity  $ID_{SM}$  and computes pseudo identity  $SID_{SM} = h(ID_{SP} || M_{SP} || RT_{SM})$ , where  $RT_{SM}$  denotes the registration timestamp of the smart meter. Further,  $SP$  chooses a temporary identity  $TIDM_{SM}$  and computes a random secret value  $PR_{SM} \in Z_g$  and finds a public key as  $PB_{SM} = PR_{SM} \cdot BS$ .

*Step-2:* The  $SP$  generates a certificate of  $SM$  as  $CRT_{SM} = M_{SP} + hfs(PB_{SP} ||$

$PB_{SM} || ) * PR_{SP} \pmod{g}$ . Further,  $SP$  stores the SM information (i.e.,  $(TIDM_{SM}, SID_{SM}, CRT_{SM}, PR_{SM}, E_n(u, v), hfs(.))$  into the memory and distributes the public key  $PB_{SM}$  for further access.

### 2.5.3 Authentication Phase

In this phase, the authentication procedures for Industrial Internet of Things (IIoT) nodes, generators, and edge servers connecting to cloud servers are discussed. Each entity in the authentication process keeps a session key, enabling secure interactions inside the framework. The authorization of each entity in the framework is developed through this method. The procedure for establishing mutual authentication and sharing session keys includes the stages below.

(i) *Authentication between SM to SM*

*Step 1:*  $SM_i$  randomly selects a number  $drn_1 \in Z_g$  and also generates a current timestamp  $CT_1$ , then evaluates  $T_1 = hfs(SID_{SM_i} || TIDM_{SM_i} || drn_1 || CT_1)$ . Then  $SM_i$  performs an encryption operation to compute  $T_2 = E_{PB_{SM_j}}(T_1)$ . Next, it  $SM_i$  evaluates the parameter  $T_3 = hfs(T_2 || CRT_{SM_i} || SID_{SM_i} || TIDM_{SM_i} || CT_1)$  and produces a request message  $M_1 = \{SID_{SM_i}, TIDM_{SM_i}, CT_1, T_2, T_3\}$  and sends it to another smart meter ( $SM_j$ ) through an open channel.

*Step-2:* After successfully receiving  $M_1$  at time  $CT_1^*$ ,  $SM_j$  verifies the timestamp to check for transmission delay using the condition  $|CT_1^* - CT_1| < \Delta T$ . If the obtained timestamp is within the valid range, then it  $SM_j$  checks the certificate using  $CRT_{SM_i}$ . The verification is done as follows:

$$BS = PB_{SP} + h(PB_{SM_i} || PB_{SP})$$

For every successful match, it  $SM_j$  fetches  $SID_{SM_i}$  corresponding data  $TIDM_{SM_i}$  from the secure database storage and computes

$$T_3^* = h(T_2 \parallel SID_{SM_i} \parallel TIDM_{SM_i} \parallel CRT_{SM_i})$$

to validate whether  $T_3^* = T_3$ . If valid, then  $SM_j$  applies decryption on  $T_2$  as

$$T_1 = D_{PR_{SM_j}}(T_2)$$

*Step-3:* Further, it  $SM_j$  randomly selects a number  $fr_1 \in Z_g$  and computes the current timestamp  $CT_2$  and generates a new temporary identity  $TIDM_{SM_i}^{new}$ , then evaluates  $SM_j = h(SID_{SM_i} \parallel SID_{SM_j} \parallel fr_1 \parallel CT_2)$  and applies encryption  $SM_1$  as  $SM_2 = E_{PB_{SM_i}}(FGS_1)$ . Next, it  $SM_j$  generates a session key  $SES_{SM_j} = hfs(TIDM_{SM_i}^{new} \parallel T_1 \parallel FGS_1 \parallel CT_1 \parallel CT_2)$ ,  $TIDM_{SM_i}^* = TIDM_{SM_i}^{new} \oplus h(SID_{SM_j} \parallel TIDM_{SM_i} \parallel CT_2)$ , and  $FG_3 = hfs(TIDM_{SM_i}^* \parallel FGS_1 \parallel CRT_{SM_j} \parallel SID_{SM_j} \parallel CT_2)$ . Then, it  $SM_j$  makes a reply message  $M_2 = \{TIDM_{SM_i}^*, FG_2, CRT_{SM_j}, SID_{SM_j}, CT_2\}$  and shares it  $SM_i$  through the open channel.

*Step 4:* After successfully receiving message  $M_2$  from  $SM_j$  at time  $CT_2^*$ ,  $SM_i$  verifies whether  $|CT_2^* - CT_2| < \Delta T$ ; if the timestamp matches and is within the range, then  $SM_i$  checks the certificate, i.e.,  $CRT_{SM_j}.BS = PB_{SP} + hfs(PB_{SM_j} \parallel PB_{SP})$ . Further, it  $SM_i$  applies decryption  $FG_2$  to get  $FGS_1 = D_{PR_{SM_i}}(FG_2)$ .

Furthermore,  $SM_i$  evaluates  $FG_3^* = hfs(TIDM_{SM_i}^* \parallel FGS_1 \parallel CRT_{SM_j} \parallel SID_{SM_j} \parallel CT_2)$  and verifies if  $FG_3^* = FG_3$ . If valid, then  $SM_i$  computes  $TIDM_{SM_i}^{new} = TIDM_{SM_i}^* \oplus h(SID_{SM_j} \parallel TIDM_{SM_i} \parallel CT_2)$  and generates a session key  $SES_{SM_i} = h(TIDM_{SM_i}^{new} \parallel T_1 \parallel FGS_1 \parallel CT_1 \parallel CT_2)$  and shares it with  $SM_j$ .

Next, it  $SM_i$  picks a current timestamp  $CT_3$  and performs session key verification  $SESV_{SM_i}$  using  $SESV_{SM_i} = h(SES_{SM_i} \parallel CT_3)$  and performs the updation of

$TIDM_{SM_i}$  and  $TIDM_{SM_i}^{new}$  in the secure database storage. Further, it  $SM_i$  generates an acknowledgment message  $M_3 = \{SESV_{SM_i}, CT_3\}$  and shares it  $SM_j$  through the open channel.

*Step 5:* After successfully receiving the acknowledgment message  $M_3$  at time  $CT_3^*$ ,  $SM_j$  checks the current timestamp using it  $|CT_3^* - CT_3| < \Delta T$  to verify the validity of the timestamp. Then,  $SM_j$  verifies  $SESV_{SM_i} = h(SESV_{SM_j} || CT_3)$ . If they match successfully, then  $SM_j$  they share the session key and perform mutual authentication  $SESV_{SM_i} (= SESV_{SM_j})$  with  $SM_i$ . Finally,  $SM_j$  updates  $TIDM_{SM_i}$  are  $TIDM_{SM_i}^{new}$  in the secure database. (ii) *SM to DRCU Authentication*

*Step 1:*  $SM_i$  randomly selects a number  $drn_1 \in \mathbb{Z}_g$  and preserves the current timestamp,  $CT_1$  and computes  $T_1 = hfs(SID_{SM_i} || TIDM_{SM_i} || drn_1 || CT_1)$ . Further, it  $SM_i$  applies encryption to  $T_1$  as  $T_2 = E_{PB_{DRCU}}(T_1)$ . Additionally, it  $SM_i$  computes  $T_3 = hfs(T_2 || CRT_{SM_i} || SID_{SM_i} || TIDM_{SM_i} || CT_1)$  and sends an access request message  $M_1 = \{SID_{SM_i}, TIDM_{SM_i}, CT_1, T_2, T_3\}$  to the respective smart meter ( $DRCU$ ) through the open channel.

*Step 2:* After successfully receiving the message  $M_1$  at time  $CT_1^*$ ,  $DRCU$  verifies the current timestamp:  $|CT_1^* - CT_1| < \Delta T$ . If the timestamp is valid, it computes  $DRCU$  and checks the certificate using  $CRT_{SM_i}.BS = PB_{SP} + h(PB_{SM_i} || PB_{SP})$ . If they match successfully,  $DRCU$  stores  $SID_{SM_i}$  the corresponding data  $TIDM_{SM_i}$  from the secure database and computes  $T_3^* = h(T_2 || SID_{SM_i} || TIDM_{SM_i} || CRT_{SM_i})$  to verify if  $T_3^* = T_3$ . If they match successfully,  $DRCU$  decrypts  $T_2$  to obtain  $T_1 = D_{PR_{DRCU}}(T_2)$ .

*Step 3:* Next,  $DRCU$  randomly selects a number  $fr_1 \in \mathbb{Z}_g$ , preserves the current timestamp  $CT_2$ , and generates a new temporary identity  $TIDM_{SM_i}^{new}$ . Then, it evaluates  $DRCU = h(SID_{SM_i} || SID_{DRCU} || fr_1 || CT_2)$  and applies encryption

$SM_1$  as  $SM_2 = E_{PB_{SM_i}}(FGS_1)$ . Next,  $DRCU$  computes a session key  $SES_{DRCU} = \text{hfs}(TIDM_{SM_i}^{new} || T_1 || FGS_1 || CT_1 || CT_2)$ ,  $TIDM_{SM_i}^* = TIDM_{SM_i}^{new} \oplus \text{h}(SID_{DRCU} || TIDM_{SM_i} || CT_2)$ , and  $FG_3 = \text{hfs}(TIDM_{SM_i}^* || FGS_1 || CRT_{DRCU} || SID_{DRCU} || CT_2)$ . Finally,  $DRCU$  generates a reply message  $M_2 = \{TIDM_{SM_i}^*, FG_2, CRT_{DRCU}, SID_{DRCU}, CT_2\}$  and sends it  $SM_i$  through the open channel.

*Step-4:* Upon successfully receiving the response message  $M_2$  from  $DRCU$  at time  $CT_2^*$ ,  $SM_i$  verifies whether  $|CT_2^* - CT_2| < \Delta T$  to confirm the validity of the timestamp. If valid,  $SM_i$  checks the certificate  $CRT_{DRCU}.BS = PB_{SP} + \text{hfs}(PB_{DRCU} || PB_{SP})$ . Further, it  $SM_i$  applies decryption  $FG_2$  to get  $FGS_1 = D_{PR_{SM_i}}(FG_2)$ . Additionally,  $SM_i$  computes  $FG_3^* = \text{hfs}(TIDM_{SM_i}^* || FGS_1 || CRT_{DRCU} || SID_{DRCU} || CT_2)$  and verifies whether  $FG_3^* = FG_3$ . If they match, it  $SM_i$  computes  $TIDM_{SM_i}^{new} = TIDM_{SM_i}^* \oplus \text{h}(SID_{DRCU} || TIDM_{SM_i} || CT_2)$  and also computes the session key  $SES_{SM_i} = \text{h}(TIDM_{SM_i}^{new} || T_1 || FGS_1 || CT_1 || CT_2)$  and sends it to  $DRCU$ . Further, it  $SM_i$  picks a current timestamp  $CT_3$  and performs computation of session key verification  $SESV_{SM_i}$  using  $SESV_{SM_i} = \text{h}(SES_{SM_i} || CT_3)$  and makes an update to the  $TIDM_{SM_i}$  and  $TIDM_{SM_i}^{new}$  in the secure database. Furthermore, it  $SM_i$  produces an acknowledgment message  $M_3 = \{SESV_{SM_i}, CT_3\}$  and shares it  $DRCU$  through an open channel.

5:- After successfully receiving the acknowledgment message  $M_3$  on time  $CT_3^*$ , then  $DRCU$  check for the timestamp using  $|CT_3^* - CT_3| < \Delta T$  to check whether the timestamp is valid or not. Further,  $DRCU$  checks  $SESV_{SM_i} = \text{h}(SESV_{DRCU} || CT_3)$ . If it matches successfully, then it  $DRCU$  shares the session key and establishes a mutual authentication  $SESV_{SM_i} (=SESV_{DRCU})$  with  $SM_i$ . Finally,  $DRCU$  make an update of  $TIDM_{SM_i}$  and  $TIDM_{SM_i}^{new}$  in the secure database.

---

**Algorithm 1** Block Commitment and Verification using Smart Contracts enabled Proof-of-Authentication (PoA) Consensus Algorithm

---

```

1: Input: Transactions ( $Tx_i$ ), public key ( $PB_{SM_i}$ ), Smart meters ( $SM_i$ )
2: Output: Block creation and verification.
3: function v(e)rify $SM_i$ ,  $Tx_i$ ,  $PB_{SM_i}$ ,  $CRT_{SM_i}$ 
4:   assert( $VERIFY(PB_{SM_i}, SMNonce[msg.sender], CRT_{SM_i})$ )
5:   assert ( $CheckAuthenticate(SES_{SM_i})$ )
6:   emit Event(msg.sender,  $Tx_i$ , ■Successful■)
7: end function
8: function ( $VERIFY$ )( $Tx_i$ , nonce,  $CRT_{SM_i}$ )
9:    $MsgSigned = msg.sender || nonce || Tx_i$ 
10:  if ( $MsgSigned, CRT_{SM_i} == OWNER$ ) then
11:    return true
12:  else
13:    return false
14:  end if
15: end function
16: function ( $CheckAuthenticate(SES_{SM_i})$ )
17:  policies =  $session_{verify}[SES_{SM_i}]$ 
18:  if ( $policies$  is successful) then
19:    A block is created with Details such as  $TX_{BLOCK} = \{Tx_i^{hash}, ID_{SM_i}, CRT_{SM_i}, PB_{SM_i}, Block_{Hash}, CT_{SM_i}, nonce\}$ 
20:    Block commit
21:  end if
22: end function

```

---

#### 2.5.4 Block verification and block creation using PoA consensus mechanism

This stage includes two different verifications of  $D_i$  using its certificate  $CRT_{D_i}$  and  $SES_{D_i}$  which enables secure authentication. The PoA consensus mechanism is applied to verify and create a particular block in the network[155]. The detailed process of authentication is mentioned in Algorithm 1 .

## 2.6 Blockchain Analysis

The result analysis of blockchain is illustrated in Figure.2.12, Figure.2.14, Figure.2.16 and 2.17. The 2.12 consists of two different analyses, i.e., transaction upload time and block mining time, which are shown in Figures.2.11 and 2.12. It can be seen that the execution time gradually increases with varying numbers of transactions and peers. Figure.2.14 illustrates block creation and block access time as shown in Figure. 2.13 and Figure.2.14. The execution time for both cases, i.e., block creation and block access, is almost close to each other. However, it can be seen that the analysis time of transaction execution increases as the SMs increase in the network. Figure.2.16 shows the non-repudiation and contract deployment time over the network with varying numbers of transactions and peers.

The non-repudiation is performed in the system so that SMs can't deny at any stage of communication in the network. The contract deployment time shows the different functionality deployed over the network. Both non-repudiation and contract deployment are shown in Figure. 2.15 and Figure. 2.16. Figure.2.17 shows the SM registrations for varying numbers of transactions.

It can be seen that the registration time increases as SMs increase in the network. The off-chain storage is performed over the IPFS distributed storage layer. The storage size for varying number of transactions are recorded in KB. It can be seen that, the storage size increasing for different set of transactions.

The analysis of SMs registration is shown in Figure. 2.17.

## 2.7 Advantages and Complexity of the Suggested Method DBSDS

1) The proposed DL-based IDS, *RENS*, is a fusion of the unsupervised VAE tech-

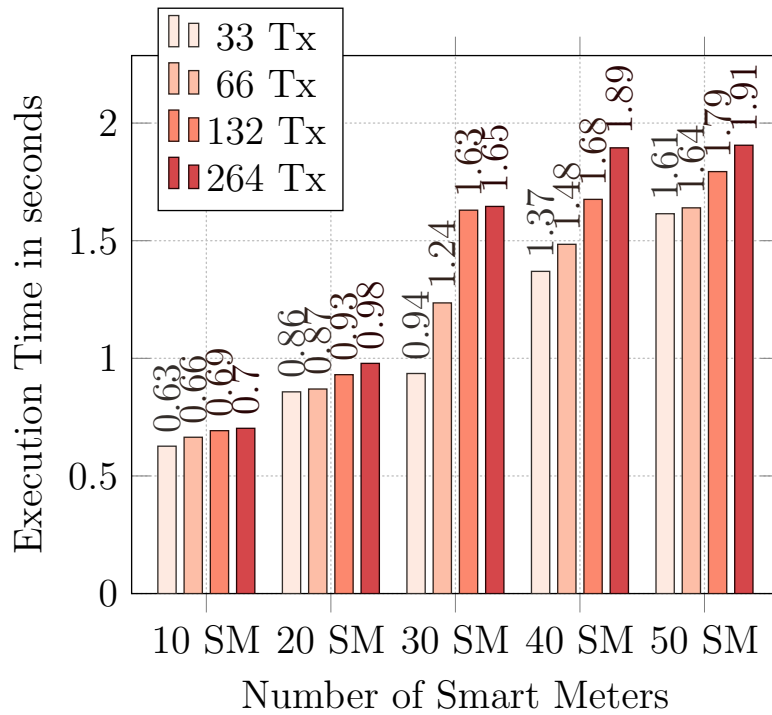


Figure 2.11: Upload time analysis of Tx

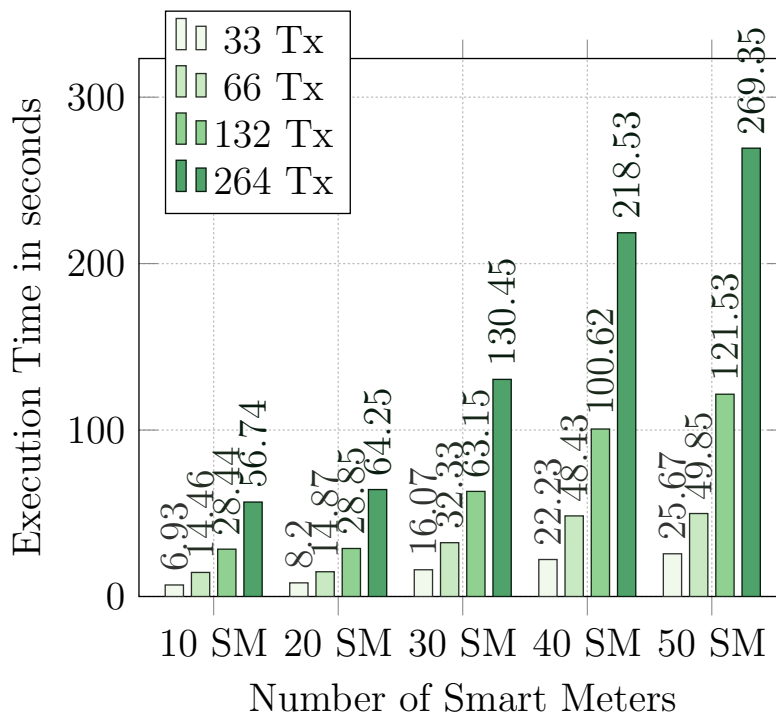


Figure 2.12: Analysis of Block Mining Time

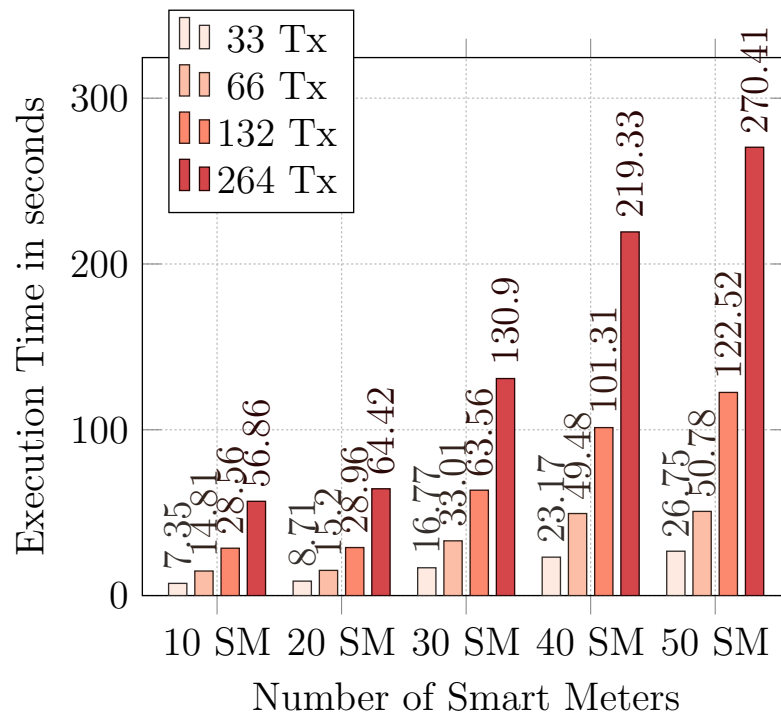


Figure 2.13: Analysis of Block Creation Time

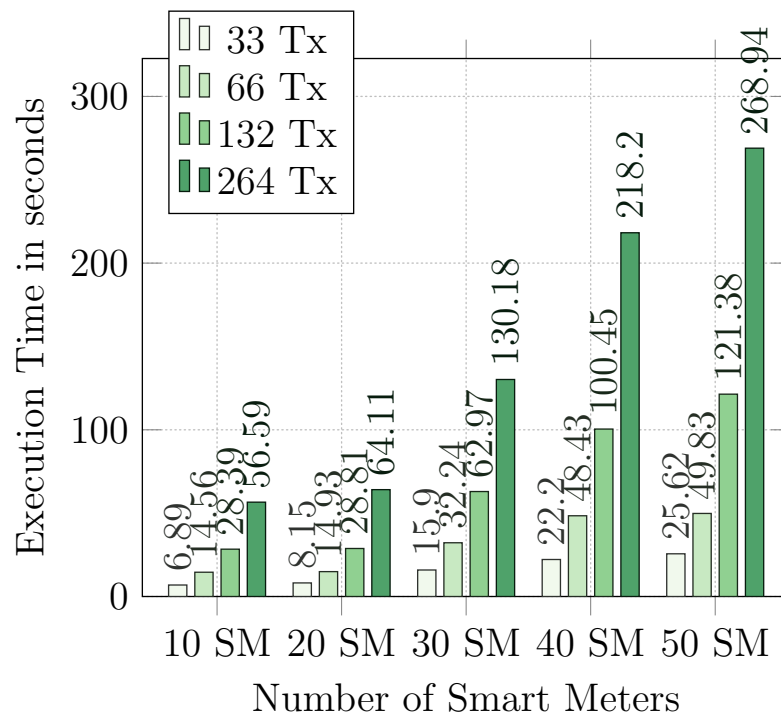


Figure 2.14: Analysis of Block Access Time

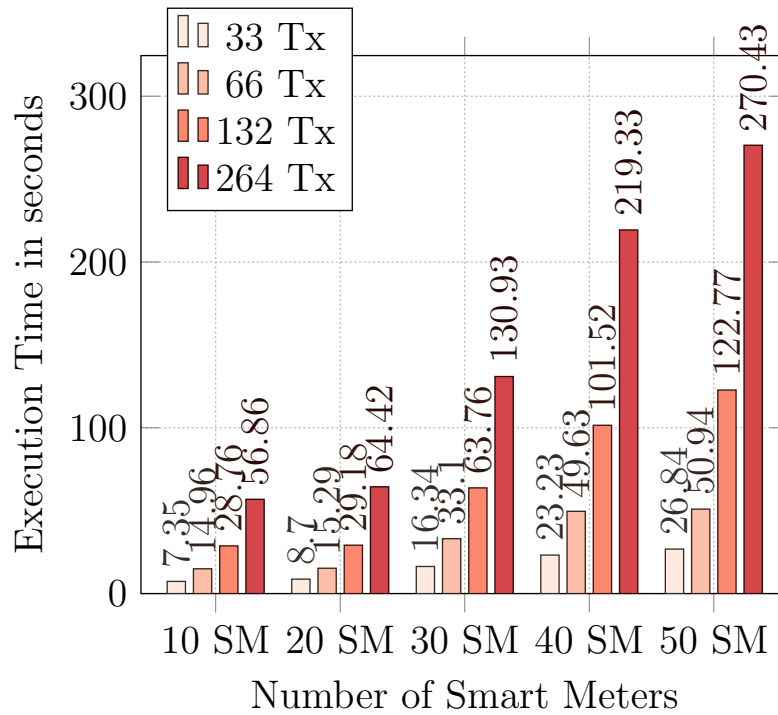


Figure 2.15: Analysis of Non-Repudiation Time

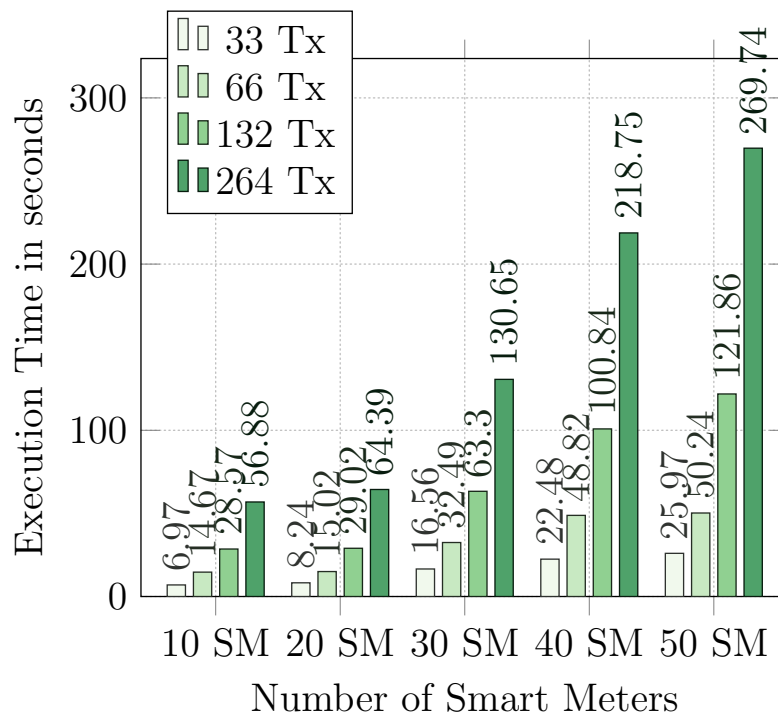


Figure 2.16: Analysis of Contract Deployment Time

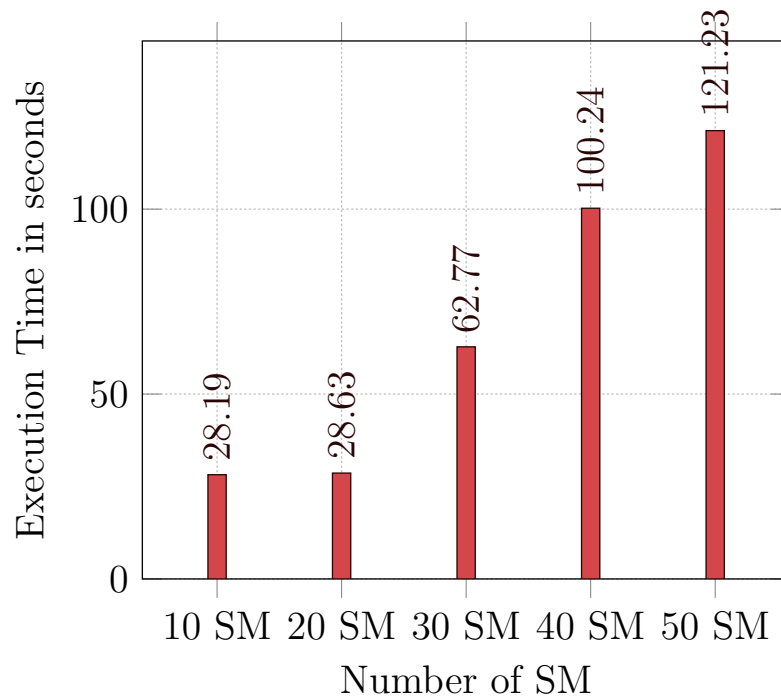


Figure 2.17: Analysis of SM Registration Time

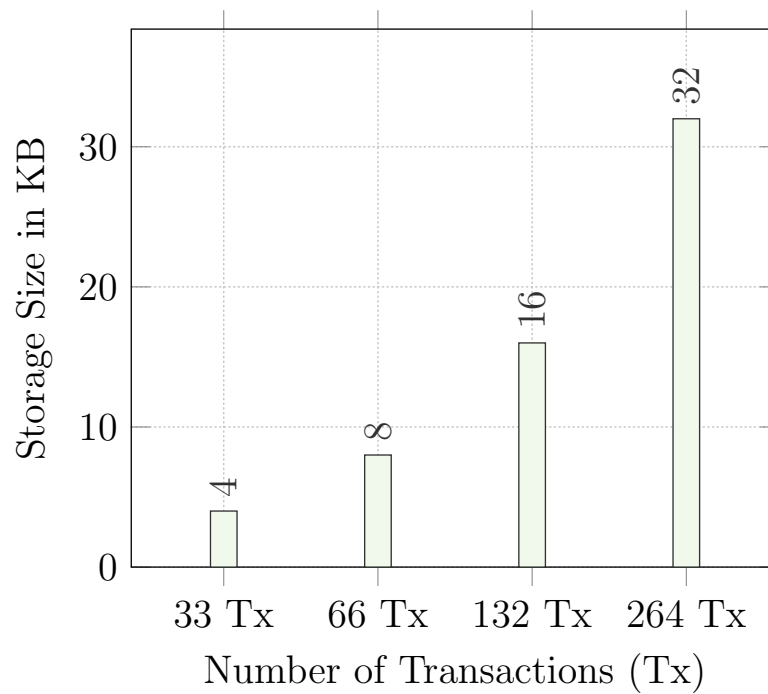


Figure 2.18: Analysis of off-chain Storage Size (in KB)

nique with ABiLSTM, which offers the highest accuracy on large, complex, and unstructured data, as can be observed from the result obtained. The complexity associated with this method is that it requires significant computational resources (e.g., GPUs), time, and energy, especially with large datasets. Performance suffers with limited or poor-quality data.

2) The security analysis and experimental results confirm the effectiveness and superiority of the proposed *DBSDS* framework, but its execution cost is higher; to run the program, gas is consumed, which is costly

## 2.8 Conclusion

This chapter successfully developed and validated the Deep Learning and Blockchain-Enabled Secure Data Sharing (DBSDS) framework, addressing the first research objective through an innovative combination of artificial intelligence and distributed ledger technology. The proposed system demonstrated superior intrusion detection capabilities by integrating RENS architectures with blockchain-based data integrity verification, achieving enhanced accuracy in identifying sophisticated cyber threats, including false data injection and man-in-the-middle attacks. The security analysis confirmed the framework's resilience against various attack vectors while maintaining computational efficiency suitable for real-time smart grid operations. Performance evaluations revealed significant improvements in detection rates and reduced false positives compared to traditional signature-based systems. The RENS framework establishes a robust foundation for secure data sharing across smart grid networks, providing the technological groundwork for extending these security principles to specialized environments such as IoT-enabled electric vehicle systems, which present unique privacy and authentication challenges addressed in Chapter 3.

## Chapter 3

# Secure and Privacy-Preserving Framework for IoT-Enabled Smart Grid Environment

### 3.1 Introduction

The secure data sharing framework established in Chapter 2 demonstrated the effectiveness of combining blockchain and deep learning technologies for general smart grid security. However, the integration of IoT-enabled Electric Vehicles (EVs) into smart grid ecosystems introduces distinct privacy and security challenges that require specialized solutions beyond the scope of traditional grid infrastructure protection. This chapter addresses the second research objective by developing a comprehensive security and privacy-preserving framework tailored specifically for IoT-enabled EV communications within Vehicle-to-Grid (V2G) environments. Building upon the blockchain-based trust mechanisms validated in previous chapters, advanced deep learning architectures, including Stacked Sparse Denoising Autoencoders (SSDAE) and Attention-based Truncated Long Short-Term Memory (ATLSTM) networks, are introduced to address the unique requirements of EV data privacy, real-time authentication, and mobile trust management[102, 106]. This framework extends our security ecosystem to encompass the dynamic, mobility-aware aspects of modern

smart grids while maintaining the integrity and efficiency principles established in the RENS system. .

### 3.2 System Models

In this section, network and threat models are discussed and analyzed in the proposed framework.

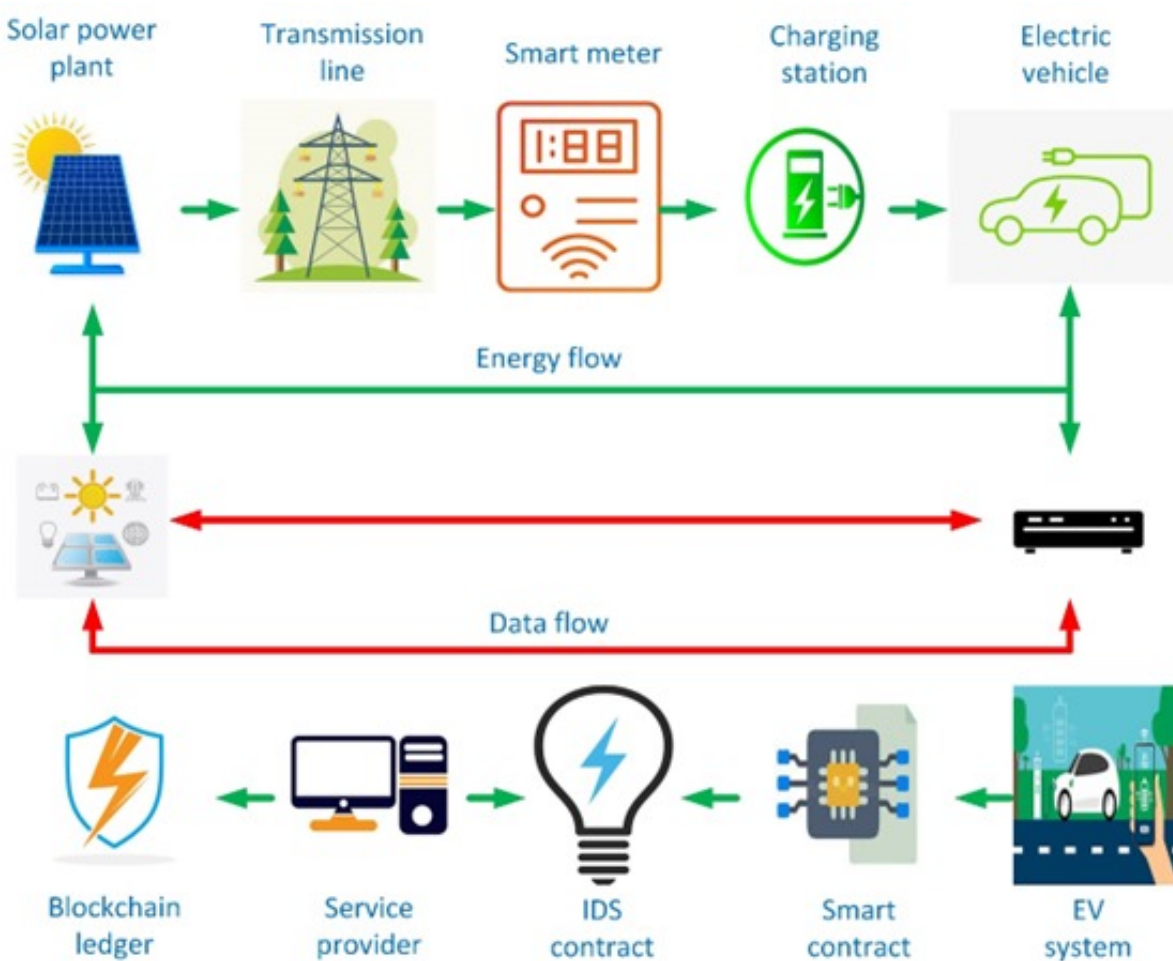


Figure 3.1: Network Model for proposed framework

### 3.2.1 Network Model

In Figure. 3.1 The network model for the proposed framework is shown. The solar power generation plant (renewable energy producer) produces electricity, which is then transmitted to various charging stations located within the city through transmission lines.

Assume that an EV needs power to charge its battery; then the entities (EV, SM, and SP) must first authenticate themselves before sharing information. Energy flow is represented by a green arrowed line, and data flow is represented by a red arrowed line in this model. Various attacks, including replay, man-in-the-middle, and impersonation attacks, are possible in the SG environment due to open wireless connectivity.

Another significant concern in the SG is the privacy of SM. An attacker can alter SM readings, causing problems with a variety of operations, including EV billing. As a result, to maintain data integrity, a secure mutual authentication and key exchange protocol between EV, SM, and SP is implemented.

After authentication has been achieved, data can be transferred between these entities using the session key. SPs, on the other hand, are often run by third-party providers, and a malicious SP can edit or leak genuine data.

As a result, the data exchanged between EV, SM, and SP has been stored using blockchain technology. Only legitimate transactions are permitted to participate in the consensus process in the proposed framework, which employs a DL-based IDS to validate the legality of the transactions sent by the SM. The suggested DL method filters aberrant transactions before sending them to a smart contract-based Proof-of-Authentication (PoA) consensus process.

The underlying technique verifies, generates, and then adds the transaction to Inter-

Planetary File System (IPFS) storage, with the cryptographic hash returned being recorded on the blockchain ledger.

As a result of our on-chain and off-chain strategies, blockchains get lighter, and block generation and access times decrease over time as schemes that store all data on the blockchain ledger are implemented[156].

### 3.2.2 Threat Model for EV and V2G Systems

In addition to conventional smart-grid cyber threats, the following EV-specific attacks are considered:

- 1) Impersonation of EV or charging station
- 2) Unauthorized charging and energy theft
- 3) False energy reporting and billing fraud
- 4) Location tracking and mobility profiling
- 5) Replay attacks during roaming
- 6) Malware infections in IoT-based chargers
- 7) Model-evasion attacks on IDS.

Adversaries may be external attackers, compromised EVs, or malicious charging stations. The Trusted Authority is assumed to be secure and uncompromised..

## 3.3 The proposed Secure and Privacy-Preserving Framework

The proposed framework consists of four integrated security layers:

- 1) Pseudonymous identity and authentication layer.
- 2) Session key management and secure communication layer.
- 3) Deep-learning-based intrusion detection layer.
- 4) Blockchain-based trusted storage and billing layer.

Only authenticated, privacy-preserved, and IDS-verified transactions are permitted to be recorded on the blockchain.

### 3.3.1 Blockchain Scheme for Secure Data Sharing

The proposed framework is divided into four phases (i) Initialization phase, (ii) the registration and verification phase, and (iii) Authentication phase, and (iv) the consensus phase for block verification and creation. The detailed working of all the phases is given below.

### 3.3.2 Privacy-Preserving System Initialization

This phase includes network setup and initialization. The trusted party ( $\mathcal{SP}$ ) initiates the setup and disseminates the functional parameters for further action in the network. These are the steps involved in the initial setup of the network mentioned below.

*Step 1:* A non-singular parameter for the elliptic curve  $ELPC_n(a, b)$  is presented in the form of  $y^2 = x^3 + ax + b \pmod{g}$  with the Galois field  $GField(g)$ , where  $g$  is denoted as a large prime number under the condition  $4a^3 + 27b^2 \not\equiv 0 \pmod{g}$  with non-singular parameters over  $\Omega$  and infinity point. Next,  $\mathcal{SP}$  selects the initial point  $IP \in ELPC_n(a, b)$  under the closest order of  $g$ , and it can be defined as  $n$  i.e.,  $n \cdot IP = \Omega$ , where  $n \cdot IP$  under the multiplicative elliptic curve point where  $n \in Z_g$ .

*Step 2:* It  $\mathcal{SP}$  uses a one-way cryptographic hash function (collision resistant)  $HF(.)$ . This operation is performed using a secure hash function (SHA-256) to generate 256 bits of message digest.

*Step-3:* The  $\mathcal{SP}$  uses an identity  $SP_{ID}$  and finds the master key  $SP_M$  and generates a random number for the private key  $SP_{PR} \in Z_g$ , where  $Z_g = \{1, 2, 3, 4, \dots, g - 1\}$ .

Table 3.1: Authentication Process between  $SM_i$  and  $SM_j$ 

| Smart Meters ( $SM_i$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | ( $SM_j$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create a random number $smdr_1 \in Z_g$<br>and timestamp $SCTS_1$<br>Compute $SL_1 = \text{HF}(SM_{PSID_i} \parallel SM_{TID_i} \parallel smdr_1 \parallel SCTS_1)$<br>$SL_2 = E_{SM_j}(SL_1)$<br>$SL_3 = \text{HF}(SL_2 \parallel SM_{CRT_i} \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SCTS_1)$<br>$SM_1 = \{SM_{PSID_i}, SM_{TID_i}, SCTS_1, SL_2, SL_3\}$<br>$\xrightarrow{\text{(secure channel)}}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Verify $ SCTS_1^* - SCTS_1  < \delta T$ , if valid<br>textit $SM_{CRT_i}.IP = SP_{PB} + h(SM_{PB_i} \parallel SP_{PB})$ , if valid<br>Fetch $SM_{PSID_i}$ corresponding data $SM_{TID_i}$ from secure database<br>Computes $SL_3^* = \text{HF}(SL_2 \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SM_{CRT_i})$<br>Verify $SL_3^* = SL_3$ , if valid<br>Decrypt $SL_1 = SM_{PR_j}(SL_2)$<br>Selects a random number $sfg_1 \in Z_g$<br>with current time stamp $SCTS_2$<br>Computes $SFG_1 = \text{HF}(SM_{PSID_i} \parallel SM_{PR_j} \parallel sfg_1 \parallel SCTS_2)$<br>Encrypt $SFG_1$ as $SFG_2 = E_{SM_{PB_i}}(SFG_1)$<br>computes a session key $SM_{SES_j} = \text{HF}(SM_{TID_i}^{new} \parallel SL_1 \parallel SFG_1 \parallel SCTS_1 \parallel SCTS_2)$<br>$SM_{TID_i}^* = SM_{TID_i}^{new} \oplus \text{HF}(SM_{PSID_j} \parallel SM_{TID_i} \parallel SCTS_2)$<br>$SFG_3 = \text{HF}(SM_{TID_i}^* \parallel SFG_1 \parallel SM_{CRT_j} \parallel SM_{PSID_j} \parallel SCTS_2)$<br>$SM_2 = SM_{TID_i}^*, SFG_2, SM_{CRT_j}, SM_{PSID_j}, SCTS_2$<br>$\xleftarrow{\text{(via open channel)}}$ |
| Verify $ SCTS_2^* - SCTS_2  < \delta T$<br>Verify if, $SM_{CRT_j}.IP = SP_{PB} + \text{HF}(SM_{PB_j} \parallel SP_{PB})$<br>Decrypt the $SFG_2$ to get $SFG_1 = D_{SM_{PR_i}}(SFG_2)$<br>Computes $SFG_3^* = \text{HF}(SM_{TID_i}^* \parallel SFG_1 \parallel SM_{CRT_j} \parallel SM_{PSID_j} \parallel SCTS_2)$<br>if $SFG_3^* = SFG_3$ valid<br>Computes $SM_{TID_i}^{new} = SM_{TID_i}^* \oplus \text{hfs}(SM_{PSID_j} \parallel SM_{TID_i} \parallel SCTS_2)$<br>Computes a session key $SM_{SES_i} = \text{HF}(SM_{TID_i}^{new} \parallel SL_1 \parallel SFG_1 \parallel SCTS_1 \parallel SCTS_2)$<br>Create current timestamp $SCTS_3$<br>Computes session key verification $SM_{ESV_i} = \text{hfs}(SM_{SES_i} \parallel SCTS_3)$<br>Updates $SM_{TID_i}$ and $SM_{TID_i}^{new}$ an insecure database<br>$SM_3 = SM_{SES_i}, SCTS_3$<br>$\xrightarrow{\text{(using open channel)}}$ | Verify $ SCTS_3^* - SCTS_3  < \delta T$<br>Verify $SM_{SES_i} = h(SM_{SES_j} \parallel SCTS_3)$ if valid<br>Updates $SM_{TID_i}$ and $SM_{TID_i}^{new}$<br>in its database securely.<br>Both $SM_i$ $SM_j$ validate the session key<br>$SM_{SES_i} (=SM_{SES_j})$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

Further, it  $\mathcal{SP}$  computes a public key  $SP_{PB} = SP_{PR} \cdot IP$ .

*Step 4:* The  $\mathcal{SP}$  preserve private key  $SP_{PR}$  and secret key  $SP_M$  and disseminates the parameters  $\{ELPC_n(a, b), IP, SP_{PB}, \text{HF}(\cdot)\}$ .

### 3.3.3 EV and Charging Station Registration with Pseudonymization

(i) *EV Registration:* The registration of EV is done by  $SP$  with the following steps mentioned below:

*Step-1:* The  $SP$  selects a unique identity  $EV_{ID}$  and computes its pseudo identity  $EV_{PSID} = h(SP_{ID} || SP_M || EV_{RT})$ , where  $EV_{RT}$  is the registration timestamp of  $EV$ . Further,  $SP$  temporary identity is created, i.e.,  $EV_{TID}$  a random number is selected to generate a private key,  $EV_{PR} \in Z_g$  and the public key  $EV_{PB} = EV_{PR} \cdot IP$ .

*Step-2:* The  $SP$  creates certificates of  $EV$  as  $EV_{CRT} = SP_M + hfs(SP_{PB} || EV_{PB} || ) * SP_{PR} \pmod{g}$ . Further,  $SP$  preserves  $EV$  credentials (i.e.,  $(EV_{TID}, EV_{PSID}, EV_{CRT}, EV_{PR}, E_n(a, b), HF(.))$  and shares the public key  $EV_{PB}$  for further access.

(ii) *SM Registration:* The registration of  $SM$  is done by  $SP$  with the following steps mentioned below:

*Step-1:* The  $SP$  selects a unique identity  $SM_{ID}$  and computes its pseudo identity  $SM_{PSID} = h(SP_{ID} || SP_M || SM_{RT})$ , where  $SM_{RT}$  is the registration timestamp of  $SM$ . Further,  $SP$  temporary identity is created, i.e.,  $SM_{TID}$  a random number is selected to generate a private key,  $SM_{PR} \in Z_g$  and the public key  $SM_{PB} = SM_{PR} \cdot IP$ .

*Step-2:* The  $SP$  creates certificates of  $EV$  as  $SM_{CRT} = SP_M + hfs(SP_{PB} || SM_{PB} || ) * SP_{PR} \pmod{g}$ . Further,  $SP$  preserves  $SM$  credentials (i.e.,  $(SM_{TID}, SM_{PSID}, SM_{CRT}, SM_{PR}, E_n(a, b), HF(.))$  and shares the public key  $SM_{PB}$  for further access.

### 3.3.4 Privacy-Aware Mutual Authentication and Key Agreement

This stage covers the authentication mechanisms among  $SM$  nodes, between  $EV$ s and  $SM$  nodes, and between  $SM$  nodes and the service provider ( $SP$ ). This authentication process is carried out using a session key for making secure communications. The whole process involves authorization of the communicating entity of the network.

Table 3.2: Authentication Process between  $SM_i$  and  $EV_j$ 

| Smart Meters ( $SM_i$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | ( $EV_j$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create a random number $smdr_1 \in Z_g$<br>and timestamp $SCTS_1$<br>Compute $SL_1 = \text{HF}(SM_{PSID_i} \parallel SM_{TID_i} \parallel smdr_1 \parallel SCTS_1)$<br>$smdr_1 \parallel SCTS_1$<br>$SL_2 = E_{EV_j}(SL_1)$<br>$SL_3 = \text{HF}(SL_2 \parallel SM_{CRT_i} \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SCTS_1)$<br>$SM_1 = \{SM_{PSID_i}, SM_{TID_i}, SCTS_1, SL_2, SL_3\}$<br>$\xrightarrow{\text{(secure channel)}}$                                                                                                                                                                                                                                                                                                                                                                                                                                             | Verify $ SCTS_1^* - SCTS_1  < \delta T$ , if valid<br>textit $SM_{CRT_i}.IP = SP_{PB} + h(SM_{PB_i} \parallel SP_{PB})$ , if valid<br>Fetch $SM_{PSID_i}$ corresponding data $SM_{TID_i}$ from secure database<br>Computes $SL_3^* = \text{HF}(SL_2 \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SM_{CRT_i})$<br>Verify $SL_3^* = SL_3$ , if valid<br>Decrypt $SL_1 = SM_{PR_j}(SL_2)$<br>Selects a random number $sfg_1 \in Z_g$<br>with current time stamp $SCTS_2$<br>Computes $SFG_1 = \text{HF}(SM_{PSID_i} \parallel EV_{PR_j} \parallel sfg_1 \parallel SCTS_2)$<br>Encrypt $SFG_1$ as $SFG_2 = E_{SM_{PB_i}}(SFG_1)$<br>computes a session key $EV_{SES_j} = \text{HF}(SM_{TID_i}^{new} \parallel SL_1 \parallel SFG_1 \parallel SCTS_1 \parallel SCTS_2)$<br>$SM_{TID_i}^* = SM_{TID_i}^{new} \oplus \text{HF}(EV_{PR_j} \parallel SM_{TID_i} \parallel SCTS_2)$<br>$SM_{TID_i} \parallel SCTS_2$<br>$SFG_3 = \text{HF}(SM_{TID_i}^* \parallel SFG_1 \parallel EV_{CRT_j} \parallel SM_{PSID_j} \parallel SCTS_2)$<br>$SM_2 = SM_{TID_i}^*, SFG_2, EV_{CRT_j}, EV_{PSID_j}, SCTS_2$<br>$\xleftarrow{\text{(via open channel)}}$ |
| Verify $ SCTS_2^* - SCTS_2  < \delta T$<br>Verify if, $EV_{CRT_j}.IP = SP_{PB} + \text{HF}(EV_{PB_j} \parallel SP_{PB})$<br>Decrypt the $SFG_2$ to get $SFG_1 = D_{SM_{PR_j}}(SFG_2)$<br>Computes $SFG_3^* = \text{HF}(SM_{TID_i}^* \parallel SFG_1 \parallel EV_{CRT_j} \parallel EV_{PSID_j} \parallel SCTS_2)$<br>if $SFG_3^* = SFG_3$ valid<br>Computes $SM_{TID_i}^{new} = SM_{TID_i}^* \oplus \text{hfs}(EV_{PSID_j} \parallel SM_{TID_i} \parallel SCTS_2)$<br>Computes a session key $SM_{SES_i} = \text{HF}(SM_{TID_i}^{new} \parallel SL_1 \parallel SFG_1 \parallel SCTS_1 \parallel SCTS_2)$<br>Create current timestamp $SCTS_3$<br>Computes session key verification $SM_{ESV_i} = \text{hfs}(SM_{SES_i} \parallel SCTS_3)$<br>Updates $SM_{TID_i}$ and $SM_{TID_i}^{new}$ an insecure database<br>$SM_3 = SM_{SES_i}, SCTS_3$<br>$\xrightarrow{\text{(using open channel)}}$ | Verify $ SCTS_3^* - SCTS_3  < \delta T$<br>Verify $SM_{SES_i} = h(EV_{SES_j} \parallel SCTS_3)$ if valid<br>Updates $SM_{TID_i}$ and $SM_{TID_i}^{new}$<br>in its database securely.<br>Both $SM_i$ $EV_j$ validate the session key<br>$SM_{SES_i} (=EV_{SES_j})$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

These are the steps that follow to verify the session key while communicating.

#### (i) SM to SM Authentication

*Step 1:*  $SM_k$  pick a random number  $smdr_1 \in Z_g$  and current timestamp  $SCTS_1$  and finds  $SL_1 = \text{HF}(SM_{PSID_i} \parallel SM_{TID_i} \parallel smdr_1 \parallel SCTS_1)$ . Next,  $SM_i$  encrypts  $SL_1$  as  $SL_2 = E_{SSM_{PB_j}}(SL_1)$ . Further, it  $SM_i$  evaluates the  $SL_3 = \text{HF}(SL_2 \parallel SM_{CRT_i} \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SCTS_1)$  and propagates the request message  $SM_1 = \{SM_{PSID_i}, SM_{TID_i}, SCTS_1, SL_2, SL_3\}$  and shares it with another smart meter ( $SM_j$ ) using

an open channel.

*Step 2:* after receiving  $SM_1$ ,  $SM_j$  verifies timestamp  $SCTS_1^* = |SCTS_1^* - SCTS_1| < \delta T$ . If it matches successfully, then it  $SM_j$  checks certificate  $SM_{CRT_i}.IP = SP_{PB} + \text{HF}(SM_{PB_i} || SP_{PB})$ . If it matches successfully, then it  $SM_j$  extracts  $SM_{PSID_i}$  the corresponding data  $SM_{TID_i}$  from the database and checks  $SL_3^* = \text{HF}(SL_2 || SM_{PSID_i} || SM_{TID_i} || SM_{CRT_i})$  and verifies  $SL_3^* = SL_3$ . If it matches successfully, then  $SM_j$  decrypt  $SL_2$  as  $SL_1 = D_{SM_{PR_j}}(SL_2)$ .

*Step 3:* Next,  $SM_j$  pick a random number  $sfg_1 \in Z_g$  and current timestamp,  $SCTS_2$  and produce a temporary identity  $SM_{TID_i}^{new}$  and finds  $SM_j = \text{HF}(SM_{PSID_i} || SM_{PSID_j} || sfg_1 || SCTS_2)$  and encrypt  $SSM_1$  as  $SSM_2 = E_{SM_{PB_i}}(SFG_1)$ . Further, it  $SM_j$  computes a session key  $SM_{SES_j} = \text{HF}(SM_{TID_i}^{new} || SL_1 || SFG_1 || SCTS_1 || SCTS_2)$ ,  $SM_{TID_i}^* = SM_{TID_i}^{new} \oplus \text{HF}(SM_{PSID_j} || SM_{TID_i} || SCTS_2)$ , and  $SFG_3 = \text{HF}(SM_{TID_i}^* || SFG_1 || SM_{CRT_j} || SM_{PSID_j} || SCTS_2)$  and generates a reply message  $SM_2 = \{SM_{TID_i}^*, SFG_2, SM_{CRT_j}, SM_{PSID_j}, SCTS_2\}$  and transmits it  $SM_i$  using an open channel.

*Step 4:* Once a message is received ( $SM_2$ ) from  $SM_j$  at time  $SCTS_2^*$ ,  $SM_i$  check whether  $|SCTS_2^* - SCTS_2| < \delta T$  is an actual timestamp or not. If it matches successfully  $SM_i$  checks certificate  $SM_{CRT_j}.IP = SP_{PB} + \text{HF}(SM_{PB_j} || SP_{PB})$ . Next,  $SM_i$  decrypts  $SFG_2$  to receive  $SFG_1 = D_{SM_{PR_i}}(SFG_2)$ . Further, it  $SM_i$  computes  $SFG_3^* = \text{HF}(SM_{TID_i}^* || SFG_1 || SM_{CRT_j} || SM_{PSID_j} || SCTS_2)$  and verifies, if  $SFG_3^* = SFG_3$  then  $SM_i$  computes  $SM_{TID_i}^{new} = SM_{TID_i}^* \oplus \text{HF}(SM_{PSID_j} || SM_{TID_i} || SCTS_2)$  and finds the session key  $SM_{SES_i} = \text{HF}(SM_{TID_i}^{new} || SL_1 || SFG_1 || SCTS_1 || SCTS_2)$  and shares  $SM_j$ . Next,  $SM_i$  pick a current timestamp  $SCTS_3$  and verify the session key  $SM_{SES_i}$  using  $SM_{SES_i} = \text{HF}(SM_{SES_i} || SCTS_3)$  and updates  $SM_{TID_i}$  and  $SM_{TID_i}^{new}$  from the database. Further, it  $SM_i$  generates an acknowl-

Table 3.3: Authentication Process between  $SM_i$  and  $SP_j$ 

| Smart Meters ( $SM_i$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | ( $SP_j$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create a random numbers $smdr_1 \in Z_g$<br>and timestamp $SCTS_1$<br>Compute $SL_1 = \text{HF}(SM_{PSID_i} \parallel SM_{TID_i} \parallel smdr_1 \parallel SCTS_1)$<br>$SL_2 = E_{SP_j}(SL_1)$<br>$SL_3 = \text{HF}(SL_2 \parallel SM_{CRT_i} \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SCTS_1)$<br>$SM_1 = \{SM_{PSID_i}, SM_{TID_i}, SCTS_1, SL_2, SL_3\}$<br>$\xrightarrow{\text{(secure channel)}}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Verify $ SCTS_1^* - SCTS_1  < \delta T$ , if valid<br>textit $SM_{CRT_i}.IP = SP_{PB} + h(SM_{PB_i} \parallel SP_{PB})$ , if valid<br>Fetch $SM_{PSID_i}$ corresponding data $SM_{TID_i}$ from secure database<br>Computes $SL_3^* = \text{HF}(SL_2 \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SM_{CRT_i})$<br>Verify $SL_3^* = SL_3$ , if valid<br>Decrypt $SL_1 = SP_{PR_j}(SL_2)$<br>Selects a random number $sfg_1 \in Z_g$<br>with current time stamp $SCTS_2$<br>Computes $SFG_1 = \text{HF}(SM_{PSID_i} \parallel SP_{PR_j} \parallel sfg_1 \parallel SCTS_2)$<br>Encrypt $SFG_1$ as $SFG_2 = E_{SM_{PB_i}}(SFG_1)$<br>computes a session key $SP_{SES_j} = \text{HF}(SM_{TID_i}^{new} \parallel SL_1 \parallel SFG_1 \parallel SCTS_1 \parallel SCTS_2)$<br>$SM_{TID_i}^* = SM_{TID_i}^{new} \oplus \text{HF}(SP_{PSID_j} \parallel SM_{TID_i} \parallel SCTS_2)$<br>$SFG_3 = \text{HF}(SM_{TID_i}^* \parallel SFG_1 \parallel SP_{CRT_j} \parallel SP_{PSID_j} \parallel SCTS_2)$<br>$SM_2 = \{SM_{TID_i}^*, SFG_2, SP_{CRT_j}, SP_{PSID_j}, SCTS_2\}$<br>$\xleftarrow{\text{(via open channel)}}$ |
| Verify $ SCTS_2^* - SCTS_2  < \delta T$<br>Verify if, $SP_{CRT_j}.IP = SP_{PB} + \text{HF}(SP_{PB_j} \parallel SP_{PB})$<br>Decrypt the $SFG_2$ to get $SFG_1 = D_{SM_{PR_j}}(SFG_2)$<br>Computes $SFG_3^* = \text{HF}(SM_{TID_i}^* \parallel SFG_1 \parallel SP_{CRT_j} \parallel SP_{PSID_j} \parallel SCTS_2)$<br>if $SFG_3^* = SFG_3$ valid<br>Computes $SM_{TID_i}^{new} = SM_{TID_i}^* \oplus \text{HF}(SP_{PSID_j} \parallel SM_{TID_i} \parallel SCTS_2)$<br>Computes a session key $SM_{SES_i} = \text{HF}(SM_{TID_i}^{new} \parallel SL_1 \parallel SFG_1 \parallel SCTS_1 \parallel SCTS_2)$<br>Create current timestamp $SCTS_3$<br>Computes session key verification $SM_{ESV_i} = \text{HF}(SM_{SES_i} \parallel SCTS_3)$<br>Updates $SM_{TID_i}$ and $SM_{TID_i}^{new}$ an insecure database<br>$SM_3 = \{SM_{SES_i}, SCTS_3\}$<br>$\xrightarrow{\text{(using open channel)}}$ | Verify $ SCTS_3^* - SCTS_3  < \delta T$<br>Verify $SM_{SES_i} = h(SP_{SES_j} \parallel SCTS_3)$ if valid<br>Updates $SM_{TID_i}$ and $SM_{TID_i}^{new}$<br>in its database securely.<br>Both $SM_i$ $SP_j$ validate the session key<br>$SM_{SES_i} (= SP_{SES_j})$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

edgment message  $SM_3 = \{SM_{SES_i}, SCTS_3\}$  and transmits it  $SM_j$  using an open channel.

*Step 5:* After receiving an acknowledgment  $SM_3$  at time  $SCTS_3^*$ , then  $SM_j$  verify if the timestamp by  $|SCTS_3^* - SCTS_3| < \delta T$  is valid or not. Next  $SM_j$  verifies  $SM_{SES_i} = \text{HF}(SM_{SES_j} \parallel SCTS_3)$ . If matched successfully,  $SM_j$  establish a session key  $SM_{SES_i} (= SM_{SES_j})$  with  $SM_i$ . Further, credential ( $SM_{TID_i}$  and  $SM_{TID_i}^{new}$ ) are updated in the database by  $SM_j$ .

(ii) *SM to EV Authentication*

*Step 1:*  $SM_k$  pick a random number  $smdr_1 \in Z_g$  and current timestamp  $SCTS_1$  and finds  $SL_1 = \text{HF}(SM_{PSID_i} \parallel SM_{TID_i} \parallel smdr_1 \parallel SCTS_1)$ . Next,  $SM_i$  encrypts  $SL_1$  as  $SL_2 = E_{SSM'PB_j}(SL_1)$ . Further, it  $SM_i$  evaluates the  $SL_3 = \text{HF}(SL_2 \parallel SM_{CRT_i} \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SCTS_1)$  and propagates the request message  $SM_1 = \{SM_{PSID_i}, SM_{TID_i}, SCTS_1, SL_2, SL_3\}$  and shares it with another smart meter ( $SM_j$ ) using an open channel.

*Step 2:* after receiving  $SM_1$ ,  $EV_j$  verifies timestamp  $SCTS_1^* = |SCTS_1^* - SCTS_1| < \delta T$ . If it matches successfully, then it  $EV_j$  checks certificate  $SM_{CRT_i}.IP = SP_{PB} + \text{HF}(SM_{PB_i} \parallel SP_{PB})$ . If it matches successfully, then it  $SM_j$  extracts  $SM_{PSID_i}$  the corresponding data  $SM_{TID_i}$  from the database and checks  $SL_3^* = \text{HF}(SL_2 \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SM_{CRT_i})$  and verifies  $SL_3^* = SL_3$ . If it matches successfully, then  $EV_j$  encrypt  $SL_2$  as  $SL_1 = D_{SM_{PB_i}}(SL_2)$ .

*Step 3:* Next,  $EV_j$  pick a random number  $sfgr_1 \in Z_g$  and current timestamp  $SCTS_2$  and produces a temporary identity  $SM_{TID_i}^{new}$  and finds  $EV_j = \text{HF}(SM_{PSID_i} \parallel SM_{PSID_j} \parallel sfgr_1 \parallel SCTS_2)$  and encrypt  $SSM_1$  as  $SSM_2 = E_{SM'PB_i}(SFG_1)$ . Further, it  $EV_j$  computes a session key  $EV_{SES_j} = \text{HF}(SM_{TID_i}^{new} \parallel SL_1 \parallel SFG_1 \parallel SCTS_1 \parallel SCTS_2)$ ,  $SM_{TID_i}^* = SM_{TID_i}^{new} \oplus \text{HF}(SM_{PSID_j} \parallel SM_{TID_i} \parallel SCTS_2)$ , and  $SFG_3 = \text{HF}(SM_{TID_i}^* \parallel SFG_1 \parallel EV_{CRT_j} \parallel EV_{PSID_j} \parallel SCTS_2)$  and generates a reply message  $SM_2 = \{SM_{TID_i}^*, SFG_2, EV_{CRT_j}, EV_{PSID_j}, SCTS_2\}$  and transmits it  $SM_i$  using an open channel.

*Step 4:* Once a message is received ( $SM_2$ ) from  $EV_j$  at time  $SCTS_2^*$ ,  $SM_i$  check whether  $|SCTS_2^* - SCTS_2| < \delta T$  is an actual timestamp or not. If it matches successfully,  $SM_i$  check certificate  $EV_{CRT_j}.IP = SP_{PB} + \text{HF}(EV_{PB_j} \parallel SP_{PB})$ . Next,  $SM_i$  decrypts  $SFG_2$  to receive  $SFG_1 = D_{SM_{PB_i}}(SFG_2)$ . Further, it  $SM_i$  com-

computes  $SFG_3^* = \text{HF}(SM_{TID_i}^* \parallel SFG_1 \parallel EV_{CRT_j} \parallel EV_{PSID_j} \parallel SCTS_2)$  and verifies, if  $SFG_3^* = SFG_3$  then  $SM_i$  computes  $SM_{TID_i}^{new} = SM_{TID_i}^* \oplus \text{HF}(EV_{PSID_j} \parallel SM_{TID_i} \parallel SCTS_2)$  and finds the session key  $SM_{SESV_i} = \text{HF}(SM_{TID_i}^{new} \parallel SL_1 \parallel SFG_1 \parallel SCTS_1 \parallel SCTS_2)$  and shares  $EV_j$ . Next,  $SM_i$  pick a current timestamp  $SCTS_3$  and verify the session key  $SM_{SESV_i}$  using  $SM_{SESV_i} = \text{HF}(SM_{SESV_i} \parallel SCTS_3)$  and updates  $SM_{TID_i}$   $SM_{TID_i}^{new}$  from the database. Further, it  $SM_i$  generates an acknowledgment message  $SM_3 = \{SM_{SESV_i}, SCTS_3\}$  and transmits it  $EV_j$  using an open channel.

*Step 5:* After receiving an acknowledgment  $SM_3$  at time  $SCTS_3^*$ , then  $EV_j$  verify if the timestamp by  $|SCTS_3^* - SCTS_3| < \delta T$  is valid or not. Next  $EV_j$  verifies  $SM_{SESV_i} = \text{HF}(EV_{SESV_j} \parallel SCTS_3)$ . If they match successfully, they  $EV_j$  establish a session key  $SM_{SESV_i}$  ( $=EV_{SESV_j}$ ) with  $SM_i$ . Further, credential ( $SM_{TID_i}$  and  $SM_{TID_i}^{new}$ ) are updated in the database by  $EV_j$ .

### (iii) SM to SP Authentication

*Step 1:*  $SM_i$  pick a random number  $smdr_1 \in Z_g$  and current timestamp  $SCTS_1$  and finds  $SL_1 = \text{HF}(SM_{PSID_i} \parallel SM_{TID_i} \parallel smdr_1 \parallel SCTS_1)$ . Next,  $SM_i$  encrypts  $SL_1$  as  $SL_2 = E_{SSM_{PB_j}}(SL_1)$ . Further, it  $SM_i$  evaluates the  $SL_3 = \text{HF}(SL_2 \parallel SM_{CRT_i} \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SCTS_1)$  and propagates the request message  $SM_1 = \{SM_{PSID_i}, SM_{TID_i}, SCTS_1, SL_2, SL_3\}$  and shares it with another smart meter ( $SM_j$ ) using an open channel.

*Step 2:* After receiving  $SM_1$ ,  $SP_j$  verifies timestamp  $SCTS_1^* = |SCTS_1^* - SCTS_1| < \delta T$ . If it matches successfully, then it  $SP_j$  checks certificate  $SM_{CRT_i}.IP = SP_{PB} + \text{HF}(SM_{PB_i} \parallel SP_{PB})$ . If it matches successfully, then it  $SP_j$  extracts  $SM_{PSID_i}$  the corresponding data  $SM_{TID_i}$  from the database and checks  $SL_3^* = \text{HF}(SL_2 \parallel SM_{PSID_i} \parallel SM_{TID_i} \parallel SM_{CRT_i})$  and verifies  $SL_3^* = SL_3$ . If it matches successfully, then  $SP_j$  encrypt  $SL_2$  as  $SL_1 = D_{SM_{PB_i}}(SL_2)$ .

*Step 3:* Next,  $SP_j$  pick a random number  $sfgr_1 \in Z_g$  and current timestamp,  $SCTS_2$  and produce a temporary identity  $SM_{TID_i}^{new}$  and finds  $SP_j = \text{HF}(SM_{PSID_i} \parallel SP_{PSID_j} \parallel Sfgr_1 \parallel SCTS_2)$  and encrypt  $SSM_1$  as  $SSM_2 = E_{SM \cdot PB_i}(SFG_1)$ . Further, it  $SP_j$  computes a session key  $SP_{SES_j} = \text{HF}(SM_{TID_i}^{new} \parallel SL_1 \parallel SFG_1 \parallel SCTS_1 \parallel SCTS_2)$ ,  $SM_{TID_i}^* = SM_{TID_i}^{new} \oplus \text{HF}(SP_{PSID_j} \parallel SM_{TID_i} \parallel SCTS_2)$ , and  $SFG_3 = \text{HF}(SM_{TID_i}^* \parallel SFG_1 \parallel SP_{CRT_j} \parallel SP_{PSID_j} \parallel SCTS_2)$  and generates a reply message  $SM_2 = \{SM_{TID_i}^*, SFG_2, SP_{CRT_j}, SP_{PSID_j}, SCTS_2\}$  and transmits it  $SM_i$  using an open channel.

*Step 4:* Once a message is received ( $SM_2$ ) from  $SP_j$  at time  $SCTS_2^*$ ,  $SM_i$  check whether  $|SCTS_2^* - SCTS_2| < \delta T$  is an actual timestamp or not. If it matches successfully  $SM_i$  checks certificate  $SP_{CRT_j}.IP = SP_{PB} + \text{HF}(SP_{PB_j} \parallel SP_{PB})$ . Next,  $SM_i$  decrypts  $SFG_2$  to receive  $SFG_1 = D_{SM_{PR_i}}(SFG_2)$ . Further, it  $SM_i$  computes  $SFG_3^* = \text{HF}(SM_{TID_i}^* \parallel SFG_1 \parallel SP_{CRT_j} \parallel SP_{PSID_j} \parallel SCTS_2)$  and verifies, if  $SFG_3^* = SFG_3$  then  $SM_i$  computes  $SM_{TID_i}^{new} = SM_{TID_i}^* \oplus \text{HF}(SP_{PSID_j} \parallel SM_{TID_i} \parallel SCTS_2)$  and finds the session key  $SM_{SES_i} = \text{HF}(SM_{TID_i}^{new} \parallel SL_1 \parallel SFG_1 \parallel SCTS_1 \parallel SCTS_2)$  and shares  $SP_j$ . Next,  $SM_i$  pick a current timestamp  $SCTS_3$  and verify the session key  $SM_{SES_{V_i}}$  using  $SM_{SES_{V_i}} = \text{HF}(SM_{SES_i} \parallel SCTS_3)$  and updates  $SM_{TID_i}$   $SM_{TID_i}^{new}$  from the database. Further, it  $SM_i$  generates an acknowledgment message  $SM_3 = \{SM_{SES_{V_i}}, SCTS_3\}$  and transmits it  $SP_j$  using an open channel.

*Step 5:* After receiving an acknowledgment  $SM_3$  at time  $SCTS_3^*$ , then  $SP_j$  verify if the timestamp by  $|SCTS_3^* - SCTS_3| < \delta T$  is valid or not. Next  $SP_j$  verifies  $SM_{SES_{V_i}} = \text{HF}(SP_{SES_{V_j}} \parallel SCTS_3)$ . If they match successfully, they  $SP_j$  establish a session key  $SM_{SES_{V_i}} (=SP_{SES_{V_j}})$  with  $SM_i$ . Further, credential ( $SM_{TID_i}$  and  $SM_{TID_i}^{new}$ ) are updated in the database by  $SP_j$ .

### 3.3.5 Consensus Mechanism for Block Verification and Creation:

This phase includes a consensus mechanism to verify and create the block over the network. The verification is performed  $\mathcal{SP}^j$  with the proper signature and timestamp. The block consists of block weight, timestamp, previous block index, current index of block, and signature. The block weight is 2 once the successful mining is done by the  $\mathcal{SP}^j$  else its value is set to 1. The entire process is discussed in Algorithm 2.

## 3.4 Deep-Learning-Based Intrusion Detection for EV Systems

The proposed deep learning scheme is mainly based on two modules: first a Stacked Sparse Denoising Autoencoder (SSDAE) for feature extraction and an Attention-based Truncated Long Short Term Memory (ATLSTM) for intrusion detection. Both modules working are explained below:

### 3.4.1 Stacked Sparse Denoising AutoEncoder (SSDAE)

The Stacked Sparse Denoising Autoencoder (SSDAE) is a variant of AutoEncoder (AE) and is an unsupervised deep learning approach. The AE mainly consists of two parts: an encoder and decoder.

The encoder is a down-sampled deterministic function that converts and expresses an input vector  $\mathcal{X}$  into a feature representation vector  $\mathcal{F}$  using a hidden pattern and a non-linearity function.

Based on nonlinearity in the training process, the decoder contains the resultant hidden vector  $\mathcal{F}$ , which is inverted to a d-dimensional vector  $\mathcal{Y}$ . Formulas (3.1) and (3.2) can be used to express the encoding and decoding processes, respectively [124].

$$\mathcal{F} = \mathcal{S}(\mathcal{W}_X * \mathcal{X} + \mathcal{B}_X) \quad (3.1)$$

---

**Algorithm 2** Block Verification and Creation using Proof-of-Authority Consensus Approach

---

```

1: Input: Set of  $\mathcal{SM}_i$ ,  $\mathcal{BLK} \leftarrow$  Block Field,  $\mathcal{PR} \leftarrow$  parent block,  $\mathcal{SM}_{ID_i} \leftarrow$  who made
   block signature,  $\mathcal{BI} \leftarrow$  Index of Block,  $\mathcal{WT} \leftarrow$  Block Weight,  $\mathcal{BP} \leftarrow$  Block commitment
   duration,  $\mathcal{MP} \leftarrow$  peer majority  $((\mathcal{SM}_i/2) + 1)$ .
2: Output: Block Verification and Creation
3: function B(l)ockSignature ( $\mathcal{SM}_i, \mathcal{T}$ ) :
4:    $\Delta \leftarrow$  Maximum  $\mathcal{SM}_i$ 
5:   Result = False for do
6:      $\mathcal{S} = \{\mathcal{T} - \Delta, \dots, \mathcal{T}\}$  do if then
7:        $\mathcal{BLK}.\text{Index} \bmod \mathcal{SM}_i == i$  then Result = True
8:     Result
9:   end function
10: function P(r)opagateBlock(): while  $t$  do
11:    $\mathcal{T} \leftarrow$  previous-block( $\mathcal{SM}_i$ ).BI
12:   wait until function BlockSignature( $\mathcal{SP}_j, \mathcal{N}$ )
13:    $\mathcal{TS} \leftarrow$  receive-timestamp( $\mathcal{SP}_j$ )
14:   wait until clock  $\geq \mathcal{TS} + \mathcal{BP}$  if  $m$  then  $\mathcal{T} + 1$ 
15:    $\mathcal{d}(\mathcal{SP}_i) == i$  then
16:      $\mathcal{BLK}.\mathcal{WT} = 2$  else
17:        $\mathcal{BLK}.\mathcal{WT} = 1$ 
18:      $\mathcal{BLK}.\mathcal{BI} = \mathcal{T} + 1$ 
19:      $\mathcal{BLK}.\mathcal{PR} = \text{PreviousBlock}(\mathcal{B}_i)$ 
20:      $\mathcal{BLK}.\mathcal{SP}_j \leftarrow \text{Signature}()$ 
21:      $\mathcal{B}_i \leftarrow \{ \mathcal{B}_i \cup \{ \mathcal{BLK} \}, \mathcal{PR}_i \cup \{ \mathcal{BLK}.\mathcal{PR} \} \}$ 
22:     Disseminate Block  $\mathcal{B}_i$ 
23:
24: end function

```

---

$$\mathcal{Y} = \mathcal{S}(\mathcal{W}_Y * \mathcal{F} + \mathcal{B}_Y) \quad (3.2)$$

where  $\mathcal{W}_X$  and  $\mathcal{W}_Y$  denotes input-to-hidden and the hidden-to-output weights. Bias of hidden and output unit is represented by  $\mathcal{B}_X$  and  $\mathcal{B}_Y$  and  $\mathcal{S}$  denotes activation function. The purpose of training AE is to reduce the number of reconstruction

errors to the absolute minimum. The loss function  $\mathcal{L}$ , which is defined as the range between  $\mathcal{X}$  and  $\mathcal{Y}$ , can be used to represent their closeness and is computed using formula (3.3)

$$\mathcal{L}_{AE} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} (x_i - y_i)^2 \quad (3.3)$$

In order to introduce sparse constraint in AE, certain neurons in the hidden layer is suppressed to increase network parameter update speed and computational efficiency, as well as simplify the representation of high-dimensional data. Let  $\mathcal{A}_{\mathcal{M}}(\mathcal{X})$  be the degree of activation of the  $\mathcal{M}th$  neuron, and is calculated using formula (3.4):

$$\hat{\rho}_{\mathcal{M}} = \frac{1}{\mathcal{N}} \mathcal{A}_{\mathcal{M}}(\mathcal{X}^{(I)}) \quad (3.4)$$

where  $\hat{\rho}_{\mathcal{M}}$  is average activation,  $\mathcal{X}^{(I)}$  denotes samples in dataset and  $\mathcal{N}$  is number of features in the dataset. Further, to ensure that the neurons in the hidden layer match the sparse criterion i.e.,

$$\hat{\rho}_{\mathcal{M}} = \rho \quad (3.5)$$

Where  $\rho$  is the sparse parameter, which is equal to zero. If the formula (3.5) does not satisfy  $\hat{\rho}_{\mathcal{M}}$ , there is need to make the following changes in formula.

$$\mathcal{KL}(\rho \parallel \hat{\rho}_{\mathcal{M}}) = \rho * \log \frac{\rho}{\hat{\rho}_{\mathcal{M}}} + (1 - \rho) * \log \frac{1 - \rho}{1 - \hat{\rho}_{\mathcal{M}}} \quad (3.6)$$

The relative entropy between parameters is denoted by  $\mathcal{KL}(\rho \parallel \hat{\rho}_{\mathcal{M}})$  and the loss of Sparse AE (SAE) can also be written as:

$$\mathcal{L}_{SAE} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} (x_i - y_i)^2 + \beta \sum_{\mathcal{N}=1} \mathcal{KL}(\rho \parallel \hat{\rho}_{\mathcal{M}}) \quad (3.7)$$

where  $\beta$  denotes weight of sparseness penalty item. Noise as an input is used to improve the robustness of the AE i.e., Denoising AE (DAE). The noised input  $\hat{X}$  is created by adding noise to the initial input  $X$  at random, and the resulting output is  $\hat{Y}$ . If  $\hat{Y}$  reproduces  $X$  to the fullest extent, the AE is robust and can write loss  $\mathcal{L}_{DAE}$  as:

$$\mathcal{L}_{DAE} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} (x_i - \hat{y}_i)^2 + \frac{\lambda}{2} (\|w_1^2\|_{\mathcal{F}}^2 + \|w_2^2\|_{\mathcal{F}}^2) \quad (3.8)$$

Finally, add sparsity and noise into an AE i.e., Sparse DAE (SDAE) using formula 3.3, 3.7 and 3.8 and can write the loss  $\mathcal{L}_{SDAE}$  as:

$$\mathcal{L}_{SDAE} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} (x_i - \mathcal{Y}_i)^2 + \beta \sum_{\mathcal{N}=1} \mathcal{KL}(\rho \parallel \hat{\rho}_{\mathcal{M}}) + \frac{\lambda}{2} (\|w_1^2\|_{\mathcal{F}}^2 + \|w_2^2\|_{\mathcal{F}}^2) \quad (3.9)$$

The SSDAE feature extractor is created by stacking SDAEs and training them in a greedy layer-wise fashion, with the hidden layer of the first AE serving as the input to the second. The resultant features are feed to ATLSTM for intrusion detection.

### 3.4.2 Attention-based Truncated Long Short Term Memory (ATLSTM)

Long Short Term Memory (LSTM) is one of the most powerful recurrent neural networks variants, capable of not only overcoming the problem of vanishing gradients but also better capturing long-term dependency in time series. LSTMs began as a memory block consisting of a memory cell and two gate control units, input gate and output gate.

The input gate protects the material stored in the memory block from irrelevant input, while the output gate protects other units from the interference of the present memory block's irrelevant information. A forgetting gate is then added to the mem-

ory block to overcome the LSTM model's inability to tolerate continuous input streams. An LSTM unit's update can be summarized as follows:

$$\mathcal{F}_T = \sigma(\mathcal{W}_f [\mathcal{H}_{T-1}; \mathcal{X}_T] + \mathcal{B}_f) \quad (3.10)$$

$$\mathcal{I}_T = \sigma(\mathcal{W}_i [\mathcal{H}_{T-1}; \mathcal{X}_T] + \mathcal{B}_i) \quad (3.11)$$

$$\mathcal{O}_T = \sigma(\mathcal{W}_o [\mathcal{H}_{T-1}; \mathcal{X}_T] + \mathcal{B}_o) \quad (3.12)$$

$$\tilde{\mathcal{S}}_T = \tanh(\mathcal{W}_s [\mathcal{H}_{T-1}; \mathcal{X}_T] + \mathcal{B}_s) \quad (3.13)$$

$$\mathcal{S}_T = \mathcal{F}_T \odot \mathcal{S}_{T-1} + \mathcal{I}_T \odot \tilde{\mathcal{S}}_T \quad (3.14)$$

$$\mathcal{H}_T = \mathcal{O}_T \odot \tanh(\mathcal{S}_T) \quad (3.15)$$

The above computations are done based on Backpropagation through time (BPTT) method that process the entire dataset sequence in both directions. However, with long sequences preserving the entire history of inputs and activation is impracticable, as in this process each gradient step delays learning.

Therefore, a truncated BPTT is used to train our proposed model. Moreover, the performance of LSTM can also get degraded as it assigns equal weight to spatial and temporal features due to the fact that the feature differences are ignored by LSTM. In order to overcome this challenge an attention mechanism has been added to LSTM output. This is done by deriving implicit representation  $\gamma_T$  using LSTM output  $\mathcal{H}_T$  via a nonlinear transformation and is written as follow:

$$\gamma_T = \tanh(\mathcal{W}_\gamma \mathcal{H}_T + \mathcal{B}_\gamma) \quad (3.16)$$

Then, based on the similarity representation  $\gamma_T$  with a context vector  $\gamma_W$ , the normalized significance weight coefficient  $\mathcal{A}_T$  is used to calculate the importance of packet vectors.  $\mathcal{W}_{TW}$  is a random initialization matrix that may focus on important data across  $\gamma_T$ . The weight coefficient for the coarse-grained properties discussed above is written as:

$$\mathcal{A}_t = \frac{\exp(\beta_t^T \beta_w)}{\sum \exp(\beta_t^T \beta_w)} \quad (3.17)$$

Finally, softmax classifier is used to classify attack types and normal class in the dataset.

### 3.5 Security and Privacy Analysis

The proposed framework undergoes a security evaluation to assess its resilience against impersonation attacks, insider threats, and man-in-the-middle attacks. The security analysis is done using formal verification. The analysis of proposed framework is discussed below.

(i) Impersonation Attack: The adversary produces temporal identification of  $SM_{TID}$  and at the same time it generates pseudo-identity  $\mathcal{SM}_{PSID}$ , to activate the operations as regular user.

However, timestamp  $\mathcal{SM}_{CTS}$  and session based approach used in the authentication process, where session key is mutually granted for the communication. This mutual grant of session includes various parameters such as signature, identification, message, and current timestamp.

Thus, doing all parameters operations in a given time interval is almost impossible, hence, this approach makes the system safe from impersonation attacks.

(ii) Insider Attack: The adversary can be insider and can theft required credential to identification of smart meters ( $\mathcal{SM}_{ID}$ ), pseudo-identity ( $\mathcal{SM}_{PSID}$ , and timestamp

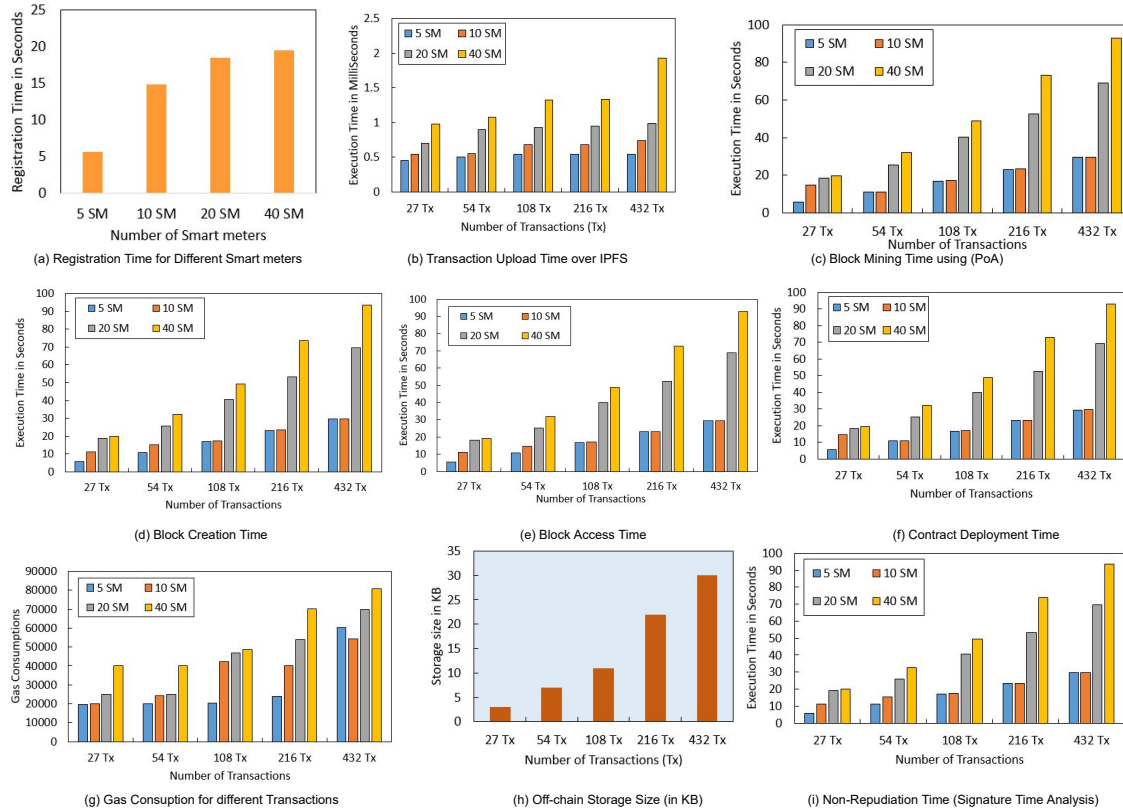


Figure 3.2: Blockchain Analysis

( $\mathcal{SM}_{CTS}$ ). But, the access of adversary can only be granted after successful verification of session based key agreement. Hence, the approach is safe from the insider attack.

(iii) MITM and Replay attack: The adversary may positively approach the  $\mathcal{SM}$  credential ( $\mathcal{SM}_{ID}$  and registration timestamp  $\mathcal{SM}_{RCTS}$ ) and can disseminate to the middle adversary for its computation and further operation in network.

But, the entire process of communication depends on session-based evaluation and performing all the credential start from  $\mathcal{SM}_{ID}$  creation to session verification. Hence, matching all the parameters in a timeframe is highly impractical and thus, the operation makes the entire framework safe from MITM attack. In the proposed framework, a privacy analysis has been conducted to assess its robustness against potential at-

tacks. Let  $\mathcal{A}$  represent an attacker who attempts to extract information using short signature techniques and modify data obtained from the secure communication channel.

To preserve privacy, the framework incorporates two key strategies:

A session-based verification scheme is employed, where a unique identifier  $SM_{ID}$  is generated for each session. This identifier acts as part of the authentication protocol, ensuring that only valid entities participate in the communication.

During the block creation process in the blockchain network, each transaction's signature is verified using the associated  $SM_{ID}$ . The verified information is then stored in the form of a transaction hash within a block. These blocks are linked sequentially, forming a secure and immutable blockchain. Thus, modifying (poisoning) the information is almost impossible as altering one block hash needs alternation is all subsequent blocks. Secondly, the proposed mode used

Stacked Sparse Denoising AutoEncoder (SSDAE) to encode the training data so that original data cannot be obtained by  $A$ . This encoding scheme rescues from inference attack. Thus, the framework satisfies the privacy-preservation successfully.

### 3.6 Performance Analysis

The proposed framework is developed on a PC with GPU of NVIDIA GeForce GTX 1060 16GB, two 2.20GHz Intel i5-3470 CPU and 64GB RAM with Win 11 operating system. The blockchain scheme is developed based on Rinkeby network. The deep learning algorithms are developed using Keras, a TensorFlow library. The SSDAE model is trained for 10 epochs using two hidden layers with 64 and 32 neurons, respectively. In contrast, the ATLSTM model is also trained for 10 epochs, utilizing two hidden layers with 32 and 16 neurons. The final layer of ATLSTM is a softmax

33 classifier, and the model employs the ReLU activation function, the Adam optimizer, and categorical cross-entropy as the loss function. Training is performed with a batch size of 100.

### 3.6.1 Datasets Description

To investigate the performance of the proposed DL-based IDS, two recent IoT-based datasets are used, called ToN-IoT [81, 83] and BoT-IoT [84, 85]. They are chosen as they were designed for a TCP/IP communication stack that accurately reflects the nature of smart grid network traffic (in particular its cloud-edge traffic).

#### ToN-IoT Dataset :

17 ToN-IoT provides a range of data sources gathered from the complete IoT system, including telemetry data from connected devices, IoT network traffic, and windows 24 and Linux operating system records. ToN-IoT was designed by the UNSW Canberra IoT[147] Labs and the Cyber Range. The ToN-IoT repository contains the network ToN-IoT dataset[81, 83]. ToN-IoT datasets were also provided in CSV format, with a tagged column indicating if the behavior was attack or normal. ToN-IoT dataset contains 1 normal and 9 attack categories, such as password attack, ransomware, data injection, backdoor, Denial of Service (DoS), Distributed Denial of Service 17 (DDoS), Cross-site Scripting (XSS), and Man-In-The-middle (MITM), and consist 209801 normal and 112929 attack instances in training and 90199 normal and 48114 attack instances in testing sets.

### BoT-IoT Dataset:

The BoT-IoT[84][85] dataset was released in 2018 by Koroniotis et al. There are almost 72 million recordings in it, with a combination of simulated and real-time settings. There are four types of attacks, including DoS, DDoS, reconnaissance and theft attack. Although DoS and DDoS packets make up a large amount of the data set. This data-set is imbalanced and consist 370 normal and 2934447 attack instances in training and 107 normal and 733598 attack instances in testing.

### 3.6.2 Evaluation Metrics

Common assessment criteria such as F-score ( $\mathcal{F} - SCORE$ ), Recall ( $\mathcal{R}$ ), Precision ( $\mathcal{P}$ ), and Accuracy ( $\mathcal{A}$ ) are used to evaluate the proposed SSDAE-ATLSTM for intrusion detection performance [150]. The recall, which is defined as a percentage of true positives divided by the total number of positive events, can be used to assess the completeness of the suggested classifier. High accuracy and recall indicate that real positives and true negatives are necessary. The F-score is the harmonic mean description of accuracy and recall. The accuracy is the poorest when the F-score is near 0, and the best when the F-score is around 1. The calculation of the suggested classifier's exactness is known as accuracy [97]. Each performance metric is explained using the formulae below.

$$\mathcal{A} = \frac{(\mathcal{TP} + \mathcal{TN})}{(\mathcal{TP} + \mathcal{FP} + \mathcal{TN} + \mathcal{FN})} \quad (3.18)$$

$$\mathcal{R} = \frac{(\mathcal{TP})}{(\mathcal{TP} + \mathcal{FN})} \quad (3.19)$$

$$\mathcal{P} = \frac{(\mathcal{TP})}{(\mathcal{TP} + \mathcal{FP})} \quad (3.20)$$

Table 3.4: Class-wise analysis (%) of proposed DL-based IDS using ToN-IoT dataset

| Parameters            | Backdoor | DDoS  | DoS   | Injection | MITM  | Normal | Password | Ransomware | Scanning | XSS   |
|-----------------------|----------|-------|-------|-----------|-------|--------|----------|------------|----------|-------|
| $\mathcal{P}$         | 99.63    | 95.04 | 96.69 | 93.68     | 96.42 | 99.99  | 98.03    | 99.26      | 99.86    | 98.39 |
| $\mathcal{R}$         | 99.80    | 92.83 | 98.87 | 95.27     | 85.98 | 99.99  | 99.22    | 99.53      | 98.01    | 97.47 |
| $\mathcal{F} - SCORE$ | 99.72    | 93.92 | 97.77 | 94.47     | 90.90 | 99.99  | 98.62    | 99.39      | 98.93    | 97.93 |

$$\mathcal{F} - SCORE = 2 * \frac{(\mathcal{P} * \mathcal{R})}{(\mathcal{P} + \mathcal{R})} \quad (3.21)$$

### 3.6.3 Blockchain Analysis

The Figure.3.2 shows blockchain analysis where different types of time analysis is computed. Figure. (3.2a) shows registration time of different SM over the network and found that as the SM increases the registration time increases accordingly. The Figure. (3.2b) shows different set of transaction upload over the IPFS storage layer. The computation time for uploading a transactions is proportionally dependent on number of transactions upload in the network. The Figure. (3.2c), Figure. (3.2d), and Figure. (3.2e) shows block mining , block creation, and block access time over the network.

It can be noticed that the computation time increases as number of transactions increases in the network.

The Figure. (3.2f) states contract deployment time over the network. It shows the computational time for different functions deployed over the network.

The Figure. (3.2g) shows the gas consumed for different set of transactions disseminated over the network by different set of SM. Figure. (3.2h) shows the off-chain storage size in KB for different set of transactions. The Figure. (3.2i) shows the different set of transactions signature by the different SM over the network. It assures the non-repudiation in the network [157],[158].

Table 3.5: Class-wise analysis (%) of proposed DL-based IDS using BoT-IoT dataset

| Parameters            | DDoS   | DoS    | Reconnaissance | Normal | Theft  |
|-----------------------|--------|--------|----------------|--------|--------|
| $\mathcal{P}$         | 99.99  | 100.00 | 99.95          | 100.00 | 82.35  |
| $\mathcal{R}$         | 100.00 | 99.99  | 99.97          | 92.52  | 100.00 |
| $\mathcal{F} - SCORE$ | 99.99  | 99.99  | 99.96          | 96.11  | 90.32  |

### 3.6.4 Deep-Learning Analysis

This subsection discuss the results obtained for the proposed IDS on various performance measurement metrics such as accuracy vs loss, class-wise analysis, confusion matrix and ROC curve. The experimental results using these metrics are discussed below:

#### Accuracy vs loss Analysis

The percentage of correctly anticipated labels in data samples is referred to as accuracy. Loss is a metric for comparing a model's output to a real-world label. Using the ToN-IoT and BoT-IoT datasets, Figure. 3.3 and Figure. 3.4 show how the proposed DL-based IDS, i.e., ATLSTM approach, improved its training and validation accuracy throughout 10 training epochs.

With the ToN-IoT dataset, it can be shown that training and validation accuracy constantly rises and converges at 98.93% and 99.11%, respectively, while ATLSTM training and validation loss steadily reduces and converges at 0.0342% and 0.0258%, respectively. The training and validation accuracy of the BoT-IoT dataset, on the other hand, improves and converges at 99.92% and 99.98%, respectively. The training and validation loss both decline progressively and converge at 0.045% and 0.041%, respectively.

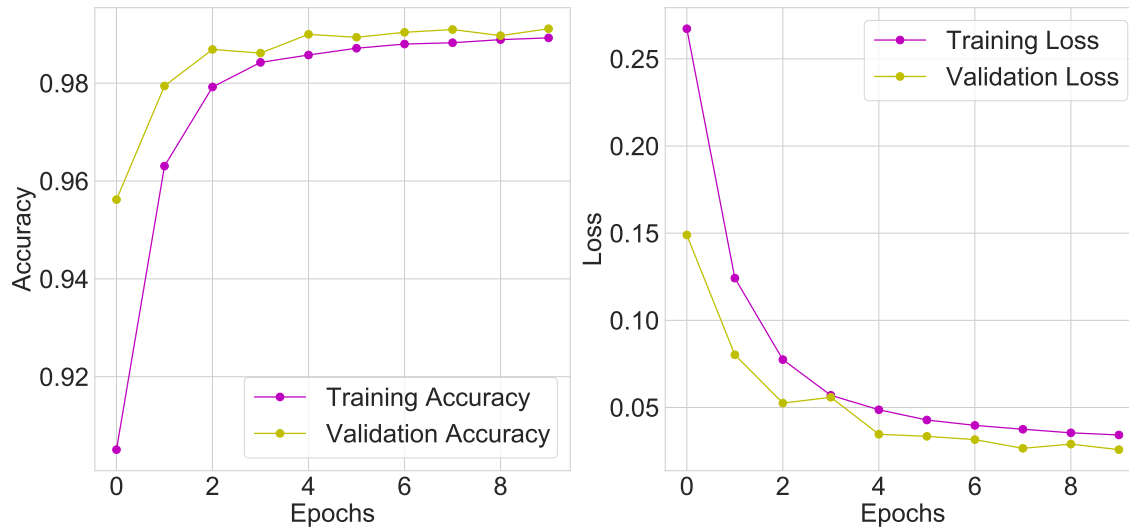


Figure 3.3: The Accuracy vs loss of proposed DL-based IDS using ToN-IoT dataset

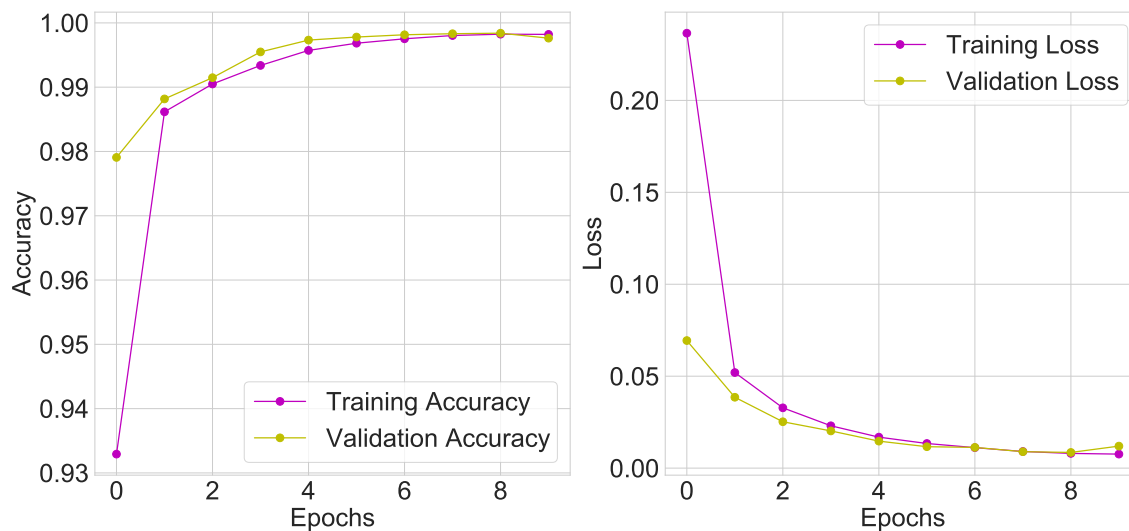


Figure 3.4: The Accuracy vs loss of proposed DL-based IDS using BoT-IoT dataset

## Class-wise Analysis

The performance of the proposed DL-based IDS is evaluated on the ToN-IoT and BoT-IoT datasets in terms of their class-wise  $P$ ,  $R$  and  $F - SCORE$ . Table 3.4 and Table 3.5 shows the performance of combined SSDAE and ATLSTM, i.e., proposed DL-based IDS on ToN-IoT and BoT-IoT datasets, respectively. It can be observed that the numerical values for these different metrics are between 92-100% with ToN-

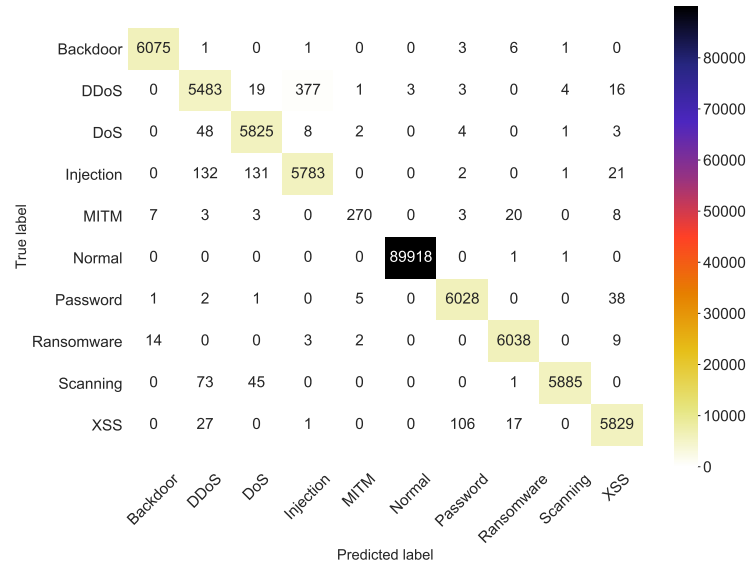


Figure 3.5: The confusion matrix of proposed DL-based IDS using ToN-IoT dataset

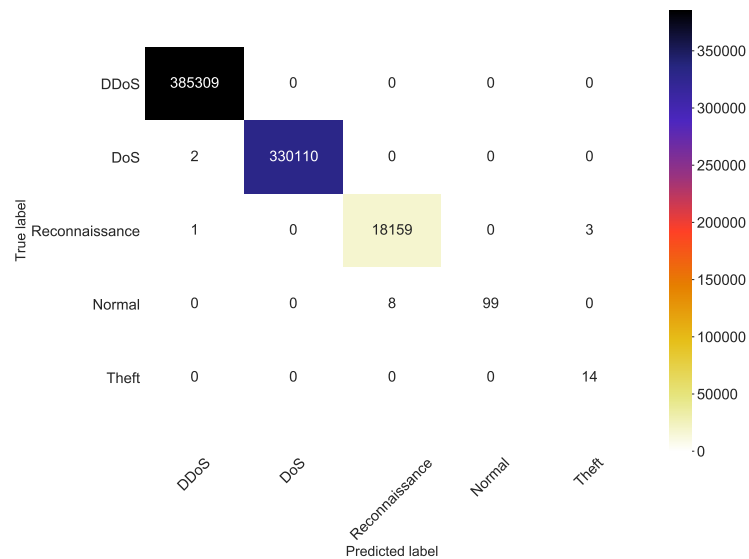


Figure 3.6: The confusion matrix of proposed DL-based IDS using BoT-IoT dataset

IoT and 82-100% with BoT-IoT dataset.

## Confusion Matrix

The confusion matrix [159] is described in classification as the summed output, which is the intrusions identified correctly or incorrectly by our suggested classifier.

This matrix contains precise information about the suggested model's performance. The true positive (without intrusion) and true negative (with intrusion) values classified by the proposed method are represented by the values in the confusion matrix's diagonal, while the false positive and false negative values of the classified data are represented by the rows and columns in the confusion matrix's other rows and columns.

Figure. 3.5 and Figure. 3.6 illustrates the confusion matrix's diagonal values with the greatest true positive and true negative values using ToN-IoT and BoT-IoT datasets, respectively.

It can be seen that the proposed DL-based has identified most of the instances present in the dataset. The figures indicate that the data harmed by various attacks are categorized more precisely with the proposed method. As a result, it can be inferred that the suggested strategy classifies huge data intrusion detection successfully.

## ROC Curve

For various threshold values, the ROC curve maps the True Positive Rate (TPR) against the False Positive Rate (FPR).

By calculating the area under the curve, the ROC curve is further measured by the Area Under the Curve (AUC). The AUC of the ideal models will achieve 1, while the random AUC classification will result in 0.5. The higher value of the AUC means that the model differentiates the output classes highly efficiently.

Figure. 3.7 and Figure. 3.8 illustrates the performance of proposed DL-based IDS in terms of ROC curves (i.e., the relationship of TPR and FPR is illustrated), before and after applying privacy-preservation technique using both datasets.

It can be seen that for all attacks and normal classes, the curves are near upper left

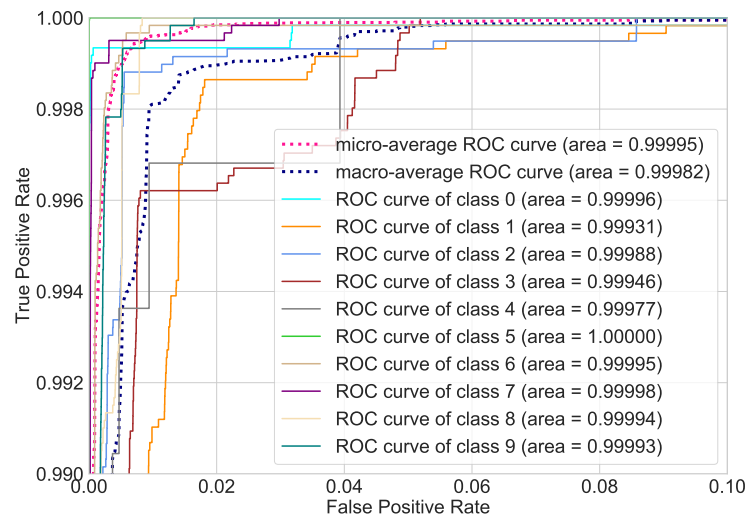


Figure 3.7: The ROC Curve of proposed DL-based IDS using ToN-IoT dataset

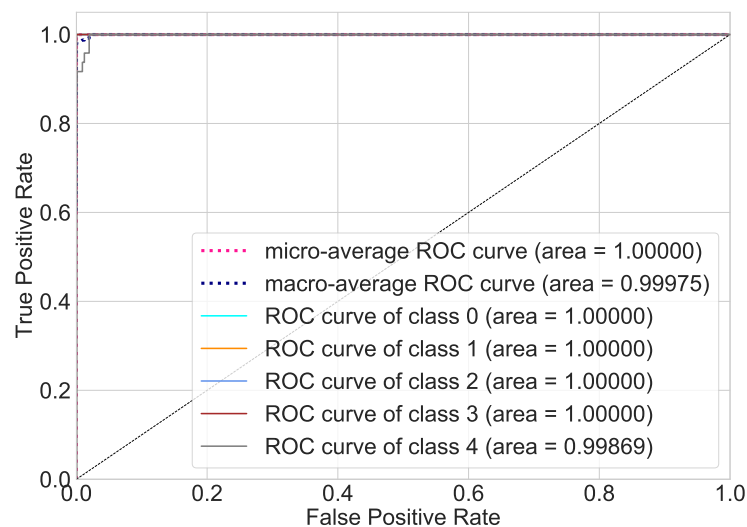


Figure 3.8: The ROC Curve of proposed DL-based IDS using BoT-IoT dataset

corner, representing perfect performance where all normal and attack instances are correctly detected.

### 3.6.5 Comparison with State-of-the-art Approaches

Finally the performance is compared of the proposed approach with some recent state-of-the-art techniques.

Table 3.6: Performance comparison with state-of-the-art intrusion detection approaches

| Authors                | Year | Approach     | Dataset | Blockchain | $\mathcal{A}$ |
|------------------------|------|--------------|---------|------------|---------------|
| Koroniotis et al. [84] | 2019 | LSTM         | BoT-IoT | No         | 98.05%        |
| Ferrag et al. [77]     | 2020 | RDTIDS       | BoT-IoT | No         | 96.95%        |
| Alsaedi et al. [80]    | 2020 | CART         | ToN-IoT | No         | 77.00%        |
| Booij et al. [81]      | 2021 | RF           | ToN-IoT | No         | 98.07%        |
| Dunn et al. [82]       | 2021 | XGBoost      | ToN-IoT | No         | 98.00%        |
| Proposed IDS           | 2022 | SSDAE-ATLSTM | ToN-IoT | Yes        | 99.14%        |
|                        |      |              | BoT-IoT |            | 99.99%        |

As shown in Table 3.6, the comparison is based on ToN-IoT and BoT-IoT datasets. The proposed DL-based IDS combines SSDAE-ATLSTM technique to design efficient and effective IDS. We have compared our approach with Alsaedi et al. [80], Booij et al. [81] and Dunn et al. [82] on ToN-IoT dataset in terms of  $\mathcal{A}$ .

It can be seen that the proposed approach has achieved higher numerical values for  $\mathcal{A}$  metric. Similarly, with BoT-IoT dataset, we have compared our performance with Koroniotis et al. [84] and Ferrag et al. [77]. It can be seen that the proposed IDS has achieved higher  $\mathcal{A}$  than these models.

### 3.7 Conclusion

This chapter successfully developed a comprehensive security and privacy-preserving framework for IoT-enabled electric vehicles, fulfilling the second research objective through innovative integration of advanced deep learning architectures with blockchain-based authentication mechanisms. The proposed system effectively addressed the unique challenges of V2G communications by implementing SSDAE for robust feature extraction and ATLSTM for sequence-aware anomaly detection, while maintaining user privacy through differential privacy techniques and pseudonymous identity management. Extensive security analysis confirmed the framework's resilience against various IoT-specific threats, including spoofing, man-in-the-middle attacks, and location privacy breaches. Performance evaluations demonstrated significant improvements in authentication efficiency and threat detection accuracy while

preserving the mobility and real-time requirements of EV ecosystems. By extending the secure data-sharing backbone of Chapter 2 with privacy and mobility awareness, this chapter completes the core cybersecurity architecture of the thesis. The next chapter further extends this architecture to metaverse- and digital-twin-assisted smart-grid security environments.

## Chapter 4

# Metaverse and Digital-Twin-Assisted Smart-Grid Security

### 4.1 Introduction

The rapid evolution of metaverse technologies and digital twins (DTs) is reshaping the way large-scale cyber–physical systems are monitored, analyzed, and controlled. In smart-grid environments, digital twins enable the creation of high-fidelity virtual replicas of physical grid components, allowing operators to observe system behavior, perform predictive analysis, test control strategies, and conduct cyber-forensics without disrupting real operations.

However, while digital twins and metaverse platforms significantly enhance situational awareness, predictive maintenance, and operator training, they also introduce new cybersecurity and data-trust challenges. The bidirectional synchronization between physical grids and their virtual counterparts enlarges the attack surface and raises critical concerns regarding the authenticity of simulation inputs, the integrity of virtual control decisions, the privacy of operational data, and trust in collaborative multi-operator environments[160].

To address these challenges, this chapter proposes a Metaverse and Digital-Twin-

Assisted Blockchain and Deep-Learning Security Framework for Smart Grids, which builds directly upon the secure backbone developed in Chapters 2 and 3. Blockchain is employed to ensure trust, auditability, and immutable synchronization, while deep learning provides real-time cyber-attack detection for both physical and virtual domains.

## 4.2 Digital Twin and Metaverse Model for Smart Grids

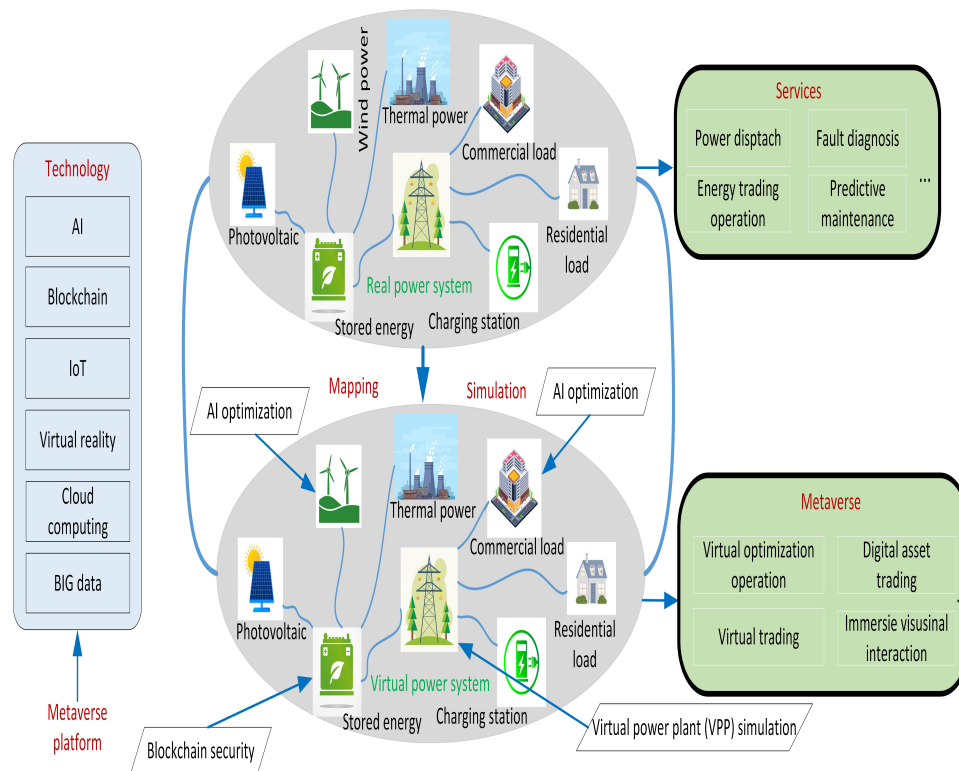


Figure 4.1: Mapping Model of Real and Virtual Power Plant

The Internet of Things (IoT), blockchain, big data, Digital Twin (DT), extended reality (XR), human-computer interaction (HCI), and other technologies have been used to map the real power system into a virtual power system. This is shown in Figure 4.1. This can make the whole process of making, sending, storing, and using electricity visible, measurable, and manageable. Parts of the mapping model are

explained below.

- 1) Real power system: It's composed of power generation, transmission, transformation, distribution, consumption, storage, and dispatching centers, which is the physical skeleton to support the construction of the power metaverse.
- 2) Virtual power system: It is the mapping of the real power system, which is the digital skeleton of the power metaverse.
- 3) Virtual interactive: It can realize perception, interaction, management, and control of real power systems and virtual power systems as the communication medium[161].
- 4) Intelligent analysis: By “virtual interactive,” the data of a real power system can be integrated into a virtual power system. Then, “intelligent analysis” can analyze the data and provide intelligent decisions in the “virtual power system,” while the decision will synchronize with the “real power system.” The data analysis decisions will also provide guidance and suggestions to users through “Human-Computer Interaction.”
- 5) Human-Computer Interaction: It is the communication medium between the Power Metaverse (PM) and users and is the key link to realize strong interaction and ubiquitous information visualization[162].

In short, each critical physical component of the smart grid—such as substations, distribution feeders, smart meters, EV charging infrastructure, and energy storage systems—is mapped to a corresponding digital twin instance within the virtual environment. The digital twin continuously receives sensor measurements, control commands, and system state updates from the physical grid via secure communication channels.

### 4.3 Metaverse-Enabled Collaborative Environment

The metaverse layer serves as a shared immersive operational space where utility operators, security analysts, maintenance engineers, and regulators can interact with the digital twin in real time. This environment enables live visualization of grid states, multi-user forensic analysis, cyberattack simulation, operator training, and decision support. However, the collaborative nature of the metaverse requires strong identity authentication, access control, and data trust guarantees, which are enforced by the blockchain-based security layer [160][162].

### 4.4 Threat Model for Digital-Twin and Metaverse Environments

In addition to traditional grid and EV cyber threats, the following DT- and Metaverse-specific attacks are considered:

Data poisoning of digital-twin inputs, false virtual control command injection, metaverse impersonation attacks, desynchronization between physical and virtual systems, model poisoning and simulation manipulation, and unauthorized access by rogue virtual avatars.

An adversary may target either the physical grid, the virtual twin, or the synchronization channel between them.

### 4.5 Blockchain application in power metaverse

Blockchain is an important way to build an identity and value system. A set of decentralized digital identity resolution technology systems is built to fix the problem that people in virtual space don't have identities and can't show who they are. Including identification, characteristics, and a way to prove who you are. A decentralized multi-certificate of identity and a trusted digital profile model are built based



#### 4.5.1 Trusted Synchronization Using Blockchain

To guarantee tamper-proof synchronization between physical and digital systems, all grid state updates are hashed and digitally signed. The transaction hash is committed to a permissioned blockchain under PoA consensus. The digital twin verifies the hash before accepting any state update.

This process ensures non-repudiation of control actions, traceability of state transitions, and protection against data injection and replay.

Only verified updates can influence virtual simulations, predictive analytics, and operator decisions inside the metaverse.

#### 4.5.2 Block Verification and Creation with the PoA Consensus Mechanism

This section provides an overview of the verification and block creation process  $\mathcal{SM}_i$  using a Proof of Authority (PoA) consensus mechanism. Within Algorithm 1, there are two distinct functions: BlockSign() and CreateBlock(). The BlockSign() function, executed by mining nodes  $\mathcal{SM}_i$ , handles the verification and signing of blocks. It checks the current timestamp  $\mathcal{CT}$  and ensures that the maximum allowed mining nodes are participating in the block verification process. A successful verification assigns a block weight of 1, while an unsuccessful one assigns a weight of 0. The CreateBlock() function considers the block weight, block signature, and mining node information. It initiates block creation upon successful authentication of miners and the relevant block information. The created block is then shared with peer mining nodes for inclusion in the current ledger. The Algorithm of POA is shown in Figure 4.3.

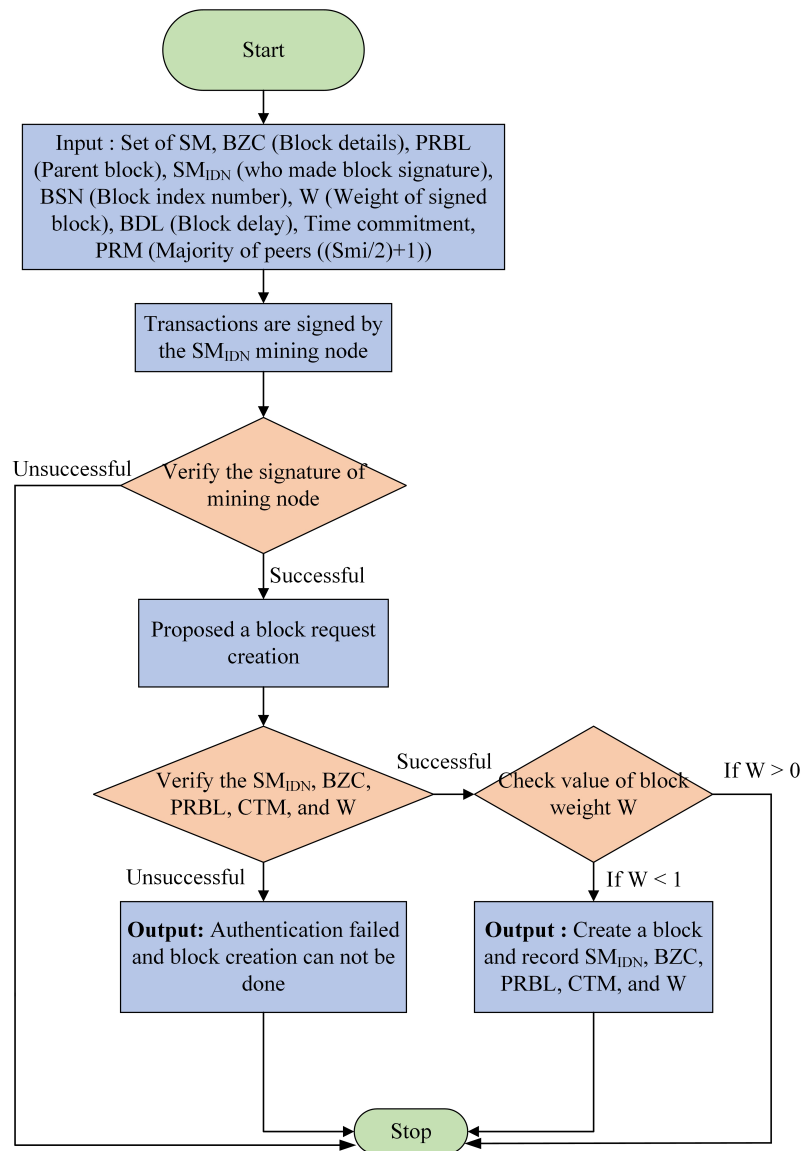


Figure 4.3: Flowchart for Block Signature and Creation using POA

## 4.6 Deep-Learning-Based Cyber-Twin Intrusion Detection

The deep-learning IDS backbone introduced in Chapter 2 is extended to operate simultaneously in physical and digital domains, forming a cyber-twin intrusion detection paradigm.

#### 4.6.1 Dual-Domain Monitoring

The IDS monitors physical network traffic, digital-twin update streams, and meta-verse session behavior.

Anomalies detected in the virtual domain (e.g., abnormal simulation commands) are correlated with physical traffic to identify stealthy attacks, insider threats, and coordinated cyber-physical intrusions.

#### 4.6.2 Real-Time Attack Response

Upon detection of malicious activity, suspicious state updates are blocked from reaching the digital twin, offending nodes are isolated, and the blockchain records the incident permanently for forensic analysis. This mechanism creates a trusted cyber-physical feedback loop between detection, and prevention.

### 4.7 Security Analysis

A security evaluation of the provided framework has been conducted to assess its resilience over impersonation attacks, insider attacks, and MITM attacks. This security analysis is carried out through formal verification, and the findings are elaborated below.

#### 4.7.1 Secure Against Impersonation Attack:

The attacker creates a temporary identification, denoted as  $SM_{TIM}$ , while simultaneously generating a pseudo-identity called  $\mathcal{SM}_{NSID}$  to engage in operations as a regular user. However, this authentication process relies on a timestamp known as  $\mathcal{SM}_{CTM}$  a session-based method, where a mutual session key is provided for the communication. This mutual granting of a session key includes parameters such as

signatures, identity, messages, and current timestamp. As a result, attempting to manipulate all of these parameters within a given time interval becomes exceedingly difficult, thus enhancing the system's security against impersonation attacks.

#### 4.7.2 Secure Against Insider Attack:

The adversary, which could potentially be an insider, may steal the necessary credentials, including the identity of the smart meter ( $\mathcal{SM}_{IDN}$ ), pseudo-identity ( $\mathcal{SM}_{NSTD}$ ), and timestamp ( $\mathcal{SM}_{CTM}$ ). However, it's important to note that the adversary's access is only permitted after the successful verification of session-based key agreement. Consequently, this approach remains secure against insider attacks.

#### 4.7.3 Secure Against MITM and Replay Attacks:

The adversary has the potential to obtain the smart meter's credentials, which include  $\mathcal{SM}_{IDN}$  the registration timestamp  $\mathcal{SM}_{RST}$ , and then pass them on to an intermediary adversary for computation and subsequent network activities. However, it's crucial to understand that the entire communication process relies on session-based assessment, encompassing all credentials starting from the creation of  $\mathcal{SM}_{IDN}$  the session to verification. Consequently, attempting to match all these parameters within a specific timeframe is highly impractical. This impracticality ensures that the operation of the entire framework remains secure against Man-in-the-Middle (MITM) attacks.

In our proposed framework, we have conducted a privacy analysis. Let's consider an attacker, denoted as  $\mathcal{A}$ , who attempts to access existing information using short signature techniques and wishes to alter the information obtained through the secure channel. Within our framework, we have incorporated two primary approaches