mathematics

MDPI

Article

AI-Empowered Attack Detection and Prevention Scheme for Smart Grid System

Aparna Kumari [1], Rushil Kaushikkumar Patel [2], Urvi Chintukumar Sukharamwala [2], Sudeep Tanwar [1,*], Maria Simona Raboaca [3,*], Aldosary Saad [4] and Amr Tolba [4]

[1] Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, Gujarat, India
[2] Computer Science and Engineering Department, R. N. G. Patel Institute of Technology, Surat 394620, Gujarat, India
[3] National Research and Development Institute for Cryogenic and Isotopic Technologies—ICSI Rm. Valcea, Uzinei Street, No. 4, P.O. Box 7 Raureni, 240050 Râmnicu Valcea, Romania
[4] Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia
* Correspondence: sudeep.tanwar@nirmauni.ac.in (S.T.); simona.raboaca@icsi.ro (M.S.R.)

Abstract: The existing grid infrastructure has already begun transforming into the next-generation cyber-physical smart grid (SG) system. This transformation has improved the grid's reliability and efficiency but has exposed severe vulnerabilities due to growing cyberattacks and threats. For example, malicious actors may be able to tamper with system readings, parameters, and energy prices and penetrate to get direct access to the data. Several works exist to handle the aforementioned issues, but they have not been fully explored. Consequently, this paper proposes an AI-ADP scheme for the SG system, which is an artificial intelligence (AI)-based attack-detection and prevention (ADP) mechanism by using a cryptography-driven recommender system to ensure data security and integrity. The proposed AI-ADP scheme is divided into two phases: (i) attack detection and (ii) attack prevention. We employed the extreme gradient-boosting (XGBoost) mechanism for attack detection and classification. It is a new ensemble learning methodology that offers many advantages over similar methods, including built-in features, etc. Then, SHA-512 is used to secure the communication that employs faster performance, allowing the transmission of more data with the same security level. The performance of the proposed AI-ADP scheme is evaluated based on various parameters, such as attack-detection accuracy, cycles used per byte, and total cycles used. The proposed AI-ADP scheme outperformed the existing approaches and obtained 99.12% accuracy, which is relatively high compared to the pre-existing methods.

Keywords: artificial intelligence; attack detection; smart grid; cryptography; cyber security; SHA-512

MSC: 68T01

check for updates

Citation: Kumari, A.; Patel, R.K.; Sukharamwala, U.C.; Tanwar, S.; Raboaca, M.S.; Saad, A.; Tolba, A. AI-Empowered Attack Detection and Prevention Scheme for Smart Grid System. *Mathematics* **2022**, *10*, 2852. https://doi.org/10.3390/math10162852

Academic Editors: Abeer Alsadoon and Luis Coelho

Received: 4 July 2022
Accepted: 8 August 2022
Published: 10 August 2022

1. Introduction

The next-generation smart grid (SG) system is one of the most prominent and extensive artificial systems in the modern era. It relies on advanced control and communication technology [1]. The existing grid infrastructure has adapted the recent advancements in several disciplines, such as artificial intelligence (AI), Internet of things (IoT), etc., in order to switch to the SG [2]. The SG embraces a complex environment of IoT-based smart sensors and actuators, secured gateways, high-performance servers, and wide-area networks to enhance the grid's efficiency, flexibility, and reliability. The grid's critical infrastructure makes it vulnerable to different hostile attacks, for instance, data modification attacks and SQL injection attacks [1,3,4].

SG is a complex infrastructure, in which any system failure due to cyberattack may lead to a huge amount of harm to the entire system in a small span of time. Therefore, early detection and prevention of cyberattacks are crucial for timely responses. However, the SG innovation comes with a risk of cyberthreats and attacks. Malicious actors (MA) may be able to manipulate smart meter (SM) readings, system parameters, energy pricing, inject data, and gain access to critical protocols, to collapse the SG in unpredictable ways [5].

The abovementioned cybersecurity issues can be addressed by adopting promising technology, i.e., AI in the SG system [3,6]. AI comprises various underlying mechanisms, for instance, machine learning (ML), reinforcement learning, etc. [7]. In SG, AI and ML are used to improve security processes and make it easier for security analysts to quickly discover, prioritize, respond to, and remediate new attacks. Several research works have been written in this regard, such as Oliveira et al.'s proposal of an intrusion-detection system. Here, the author evaluated the performance of various ML models such as random forest (RF), multi-layer perceptron (MLP), and long-short term memory (LSTM) for intrusion detection [8]. Then, Farrukh et al. proposed a two-layer hierarchical ML model with an accuracy of 95.44% for attack detection in an SG system [9]. Next, authentication systems and biometric-based authentication schemes are deployed in the SG environment by the Khan et al. [10]. In [11], a false data injection attack-detection mechanism is presented by using the Kalman filter technique. Then, an advanced attack classification approach is presented with extreme gradient boosting (XGBoost) for SG [6]. Next, a reinforcement learning-based online cyberattack detection approach is conferred [12].

Due to cybersecurity vulnerabilities, energy theft is one of the most prevalent challenges in the SG, and it has long been a source of concern for utility companies. Because all advanced cyberattack tools and manufacturers' technical specifications for common system-control equipment are available on the Internet and require no technical knowledge to operate, it becomes easy for MA to target SG. As a result, cybersecurity has become a top priority for SG, demanding a significant investment of research focused on its detection and prevention [4]. Apart from security issues, it also has several other issues like trust and privacy issues [13].

Preventing the SG system from getting attacked is one of the crucial tasks, as any small attack jeopardizes the entire system. Hence, cryptography is one of the prominent solutions to avoid attacks in the first place. Cryptography has several advantages over similar technologies, including confidentiality, authentication, data integrity, and non-repudiation [14]. Cryptographic mechanisms are critical for maintaining security and privacy in an SG communication system [15]. Among various cryptography mechanisms, elliptic curve cryptography (ECC) is the future technology for SG because it has a faster performance, allowing for more data to be sent with the same level of security [16]. Additionally, a recommender system can be connected with SG to strengthen it. Rezaimehr et al. [17] analysed 25 research samples from 2009 to 2019 on the collaborative filtering recommender system (CFRS) for attack detection. Then, Rubio et al. [18] designed a recommender system for privacy-preserving solutions in smart metering. Next, Patel et al. [19] presented an AI-empowered recommender system for harvesting renewable energy.

Many researchers have contributed to the security of SG; for example, Yan et al. in [20] suggested a holistic approach for SG security which includes existing mechanisms like public key infrastructure (PKI), authentication mechanisms, and trusted computing elements as per the industry standards. Next, Liu et al. [21] provides direction to future research work for the integration of disciplines for cybersecurity and privacy issues in the SG system [21]. Then, Wang et al. [22] includes more extensive use case studies to examine potential security vulnerabilities in subsystems of the grid. Moreover, various cyberattack-detection mechanisms are available, although the early detection with prevention has not been fully exploited [23]. Therefore, this paper proposes an AI-ADP scheme, i.e., an AI-based attack detection and prevention mechanism using cryptography in the SG system. Here, AI-ADP performs AI-based attack detection by using the XGBoost mechanism. Then, we employed a powerful Secure Hash Algorithm-512 (SHA-512) to ensure data integrity and improve SG system security. SHA-512 belongs to the SHA-2 family of cryptography algorithms. It uses a hashing function to generate the hash value that it receives. SHA-512 has proved its substantial performance gain over SHA-256 due to the doubled input block size [24]. Table 1 shows all the nomenclature used in this research work.

Table 1. Nomenclature table.

Symbol	**Description**
$\overline{\gamma}_i^{(t)}$	The predicted data value of ith sample
γ_i	The actual data value of ith sample
η_i	Features information of the ith sample, $x_i \in$ dataset
$\sum_{i=1}^{n} \Lambda(\overline{\gamma}_i^{(t-1)}, \gamma_i)$	The loss function of the ith sample
$\Delta(F_t)$	Regular term of objective function to prevent over-fitting
$\varrho_{t(\eta_i)}$	Shows result of the decision tree
T	No. of leaf nodes of Tree
δ	Contraction coefficient of T
N	Total number of area with smart meters.
m	Total number of residential houses with smart meters in each area.
$thres$	threshold value of m
$\mathfrak{ED}$	Energy consumption Data
n	Total number of energy data
F_t	Represents the tth decision tree
A_i	First derivative of loss function
Γ_i	Second derivative of loss function
κ	Sum of first derivatives after splitting of node
φ	Sum of second derivatives after node splitting
λ	penalty coefficient of the score of leaf node $\varrho_{t(\eta_i)}$
Δ	coefficient of the regularization term
L	Left node
R	Right node
NC	Minimum number of characters in a data file

1.1. Motivation

The motivation of this research work is elaborated as follows.

- The state-of-the-art attack-detection approaches have various security concerns, like data-modification attacks, data integrity, and many more. They have focused on cyberattack detection by using conventional algorithms. However, it can easily be compromised by using a high-end computational system. Thus, there is a need for a mechanism that detects the attacks with very high accuracy and also prevents the SG system from being jeopardized.
- The existing approaches such as that by Bansal et al. [25] discussed various ML techniques for identifying distinct DoS attack types but didn't consider the severe impact of data modification and SQL injection attack in the SG environment. Therefore, there is a requirement for detection and a prevention mechanism for the SG system.
- The nonmalicious data flow within the SG network is still strained from the data-modification attacks using modern computing capabilities. An attacker can target a particular sender to intercept the message request and compromise the entire SG

system. Therefore, there is a need for strong cryptography technology like SHA-512 that securely stores data at the receiver and sender end to ensure data integrity.

- Motivated by this, we have explored the integration of AI-empowered approaches like XGBoost and SHA-512 to secure the data communication with the SG system and evaluate the performance of the proposed scheme.

1.2. Research Contributions

Following are the research contributions of this paper.

- This paper proposes an AI-empowered XGBoost method to detect cyberattacks based on binary classification problems in the SG system. The proposed model employed a nonlinear method in place of the traditional linear XGBoost method and used Taylor expansion with second-order approximation.
- We designed a data-integrity and attack-prevention algorithm for the SG system by including SHA-512 cryptography. This enables one to hash the data and lower the risk of data-manipulation attacks.
- We evaluated the performance of the proposed AI-ADP scheme by comparing it with the preexisting approaches based on various parameters such as detection accuracy, cycles used per byte, and total cycles used for hashing.

1.3. Organization of the Paper

The rest of the paper is organized as follows. Section 2 presents state-of-the-art approaches with comparative analysis. Next, Section 3 discusses the system model and problem formulation of the proposed AI-ADP scheme. Then, Section 4 presents the details of the workflow of the AI-ADP scheme. In Section 5, experimental results are emphasized to prove the effectiveness of the proposed scheme. Next, Section 6 presents the discussion section, which shows the key findings in this research work and compares the findings with the findings of recent works. Lastly, Section 7 presents concluding remarks with future scope. Figure 1 shows the flow diagram of the entire manuscript with individual division of the proposed AI-ADP scheme.

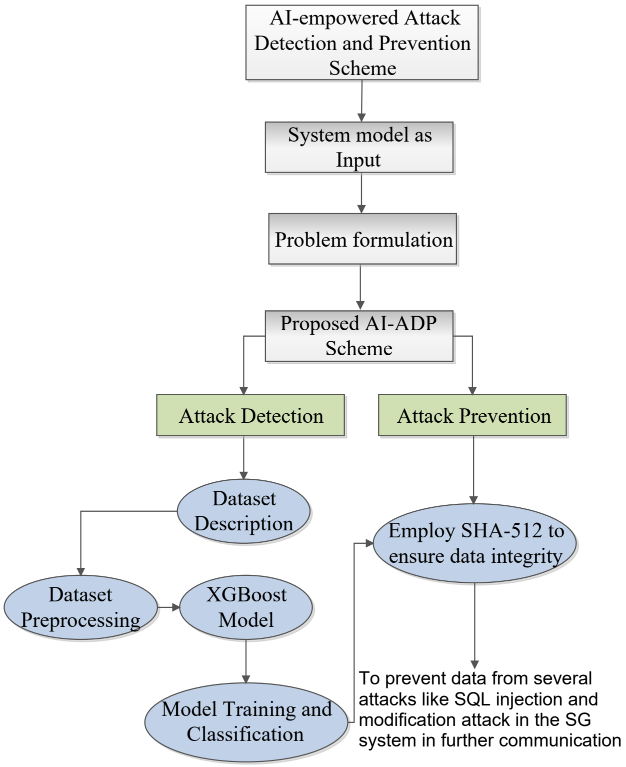

AI-empowered Attack Detection and Prevention Scheme
System model as Input
Problem formulation
Proposed AI-ADP Scheme
Attack Detection
Attack Prevention
Dataset Description
Dataset Preprocessing
XGBoost Model
Model Training and Classification
Employ SHA-512 to ensure data integrity
To prevent data from several attacks like SQL injection and modification attack in the SG system in further communication

Figure 1. Flow diagram of the proposed AI-ADP scheme.

2. Related Work

This section highlights the AI-driven state-of-the-art approaches with their advantages and disadvantages. Then, a comparative analysis of the existing research works with the proposed scheme is presented.

In [12], Kurt et al. proposed a reinforcement learning-based approach for the online detection of attacks. The proposed approach reduced the number of false alarms compared to similar existing technologies. In this approach, the model only supports a single agent that is required to be expanded to support multiple agents. Next, Morstyn et al. [26] developed the XGboost model to forecast global solar radiation (GSR) by using temperature and precipitation in climates. In this case study, the developed model is compared with the support vector machine (SVM)-based model, which the developed model outperformed. In this study, the authors used just maximum and minimum values instead of actual values for GSR predictions. Next, Cherif et al. [27] proposed an XGboost-based model to classify home network traffic. When tested on a dataset with real flows, the model had an accuracy of 99.5%. However, the proposed model requires refinement in terms of the classification of online traffic. Then, Camana et al. [28] presented an ML-based approach to detect attacks on SG. The system's advantages include its execution time, which is the shortest when compared to other state-of-the-art techniques. There is a possibility of data loss that may occur as an ML-based dimension reduction is considered.

Su et al. [29] proposed a model for identifying dynamic load-balancing attacks in the SG system. This model took the shortest time to execute, but the system's flaw is that there is no empirical basis to select vulnerable loads or other power system characteristics. In order to handle this, Patnaik et al. [30] proposed an XGboost-based classifier. This presented approach is inefficient when working with sparse and unstructured data. Khamaiseh et al. [31] developed a novel adversarial testing strategy for denial of service (DoS) [32] attack-detection systems. Then, Zivkovic et al. [33] used the firefly algorithm to boost the effectiveness of the XGBoost classifier based on hyperparameters for network-intrusion detection. In this network-intrusion detection system, the proposed approach minimizes the rate of false positives and false negatives. Table 2 presents a comparative analysis between the proposed scheme and the existing approaches from 2018 until the present. Next, the scope of this research work includes theoretical and experimental existing work from 2010 to the present. The challenges related to the cyberattacks in the SG system are discussed thoroughly. To the best of our knowledge, most of the works have highlighted the application of AI in the SG system to detect attacks. The integration of attack detection and prevention by using SHA-512 is not fully explored yet. The proposed AI-ADP scheme mitigates the furthermost cyberattack issues like energy information disclosure, data-modification attack, etc.

Several works exist on attack detection but very few incorporate the AI-based approach for detection of attacks and prevention mechanisms [34,35]. In this paper, we proposed an AI-based scheme, i.e., the AI-ADP scheme that incorporated AI for attack detection in the SG system. Then, to prevent the attack, the SHA-512 is encompassed to ensure data integrity in the SG system. The detailed discussion of the AI-ADP scheme is presented in the subsequent section.

Table 2. A comparative analysis of the proposed AI-ADP scheme with the existing approaches.

Approaches	Year	Short Description	Merits	Demerits
Kurt et al. [12]	2018	An RL-based solution for online detection of attacks.	Number of false alarms reduced compared to the existing approaches.	The presented solution needs to be expanded for multiple agents.
Morstyn et al. [26]	2018	Developed XGboost model to forecast global solar radiation (GSR)	Temperature and precipitation in climates are included in this solution	For GSR forecasting, the authors employed only maximum/minimum values rather than the actual value.
Bansal et al. [25]	2018	Various ML techniques for identifying distinct denial-of-service (DoS) attack types is discussed	Parameter tuning algorithm is highlighted for better performance.	The impact of data modification and SQL injection attacks in SG environment is not discussed.
Cherif et al. [27]	2019	Proposed an XGBoost model for home network traffic classification.	On a dataset with real flows, the proposed model achieved 99.5% accuracy.	Proposed model needs to be improved for online traffic.
Camana et al. [28]	2020	Proposed a dimension reduction-based ML algorithm to detect attack on SG.	The proposed approach takes the shortest execution time.	Data loss may arise as a result of dimension reduction.
Alqahtani et al. [36]	2020	A successful and efficient method for detecting IoT botnet attacks using extreme gradient boosting (GXGBoost) model.	The GXGBoost model and the Fisher-score-based feature selection method is used for attack detection.	Prevention mechanism for data modification and SQL injection attacks need to be discussed.
Su et al. [29]	2021	A dynamic load altering attack detection model for SG.	The proposed approach takes the shortest execution time.	There is no empirical foundation for selecting susceptible loads or other power system characteristics.
Patnaik et al. [30]	2021	Presented a XGboost based classifier.	By using XGboost, over-fitting can be reduced.	For sparse and unstructured data presented method need to be improved.
Khamaiseh et al. [31]	2021	A novel ML-based approach for DoS attack detection.	Better performance compare to similar approaches.	Emphasis only DoS, other attacks required to be included in this approach.
Zivkovic et al. [33]	2022	Optimize XGBoost classifier for network intrusion detection.	Minimized false positives and false negatives in network intrusion detection systems.	Attack detection accuracy required to be improved.
The proposed AI-ADP scheme	2022	The proposed scheme performs AI-based attack detection and prevention in SG system.	It obtained 99.12% accuracy while detecting attacks by using XGBoost and SHA-512 is incorporated to improve SG system security.	-

3. System Model and Problem Formulation

This section presents the formulated problem and system model of the proposed AI-ADP scheme.

3.1. System Model

Figure 2 shows the system model for the AI-ADP scheme, which consists of three entities (E), i.e., end-consumer (EC), electric utility companies (EUC), and attacks ($\mathbb{A}$) by intruders. In the proposed scheme, the end-customer consumes energy, generating energy-consumption data through smart meter (SM). SM communicates with the IoT cloud servers through available home routers. The energy-consumption data ($\mathfrak{ED}$) is collected by EUC for processing and analysis. During the collection of data, various attacks, i.e., $\{\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_3, \ldots, \mathbb{A}_m\} \in \mathbb{A}$ can modify $\mathfrak{ED}$ using several techniques like data-modification attacks, SQL injection attacks, etc.

To handle the aforementioned issues, the proposed AI-ADP scheme uses the XGBoost mechanism to detect the attacks on $\mathfrak{ED}$ and classify them in binary classification, i.e., attacked and non-attacked data. The proposed AI-ADP scheme discussed several attacks like SQL injection, data-modification attacks, and data-fabrication attacks through invaded SM in residential houses. Next, even low-risk attacks in the SG system jeopardize the entire system and raise questions about the quality of services. Therefore, to handle the data-security issue, it is essential to prevent it despite handling it after getting attacked. Hence, the proposed AI-ADP scheme employs hashing mechanism, i.e., SHA-512, to perform secure communication whenever data is transferred or shared among stakeholders such as SG administrator, *EC*, and *EUC*. With the help of SHA-512, the proposed scheme ensures that the receiver end has the same data transmitted from the source end. If there are any discrepancies, then the proposed scheme generates the notification message that easily handles attacks like SQL injection and data-modification attacks as soon as it impacts the SG system.

XGBoost is a relatively modern ensemble-learning technique that is gaining momentum in real-world applications such as intelligent transportation, smart health-care, etc. [6]. XGBoost uses gradient boosting to obtain the best objective values for the best decision. It is one of the most scalable, reliable, and enhanced variants of the gradient-boosting algorithm that focuses on efficacy, computing speed, and model performance. Next, to prevent the attacks, the SHA-512-based cryptography approach is employed to secure the data communication. A recommender system is employed that suggests SHA-512 for data file hashing because implementing SHA-512 requires a table of eighty 64-bit constants (a 640 bytes lookup table). As storing data for the lookup table can be extremely expensive for small amounts of data (small files), this recommender system only recommends using it for large data files.

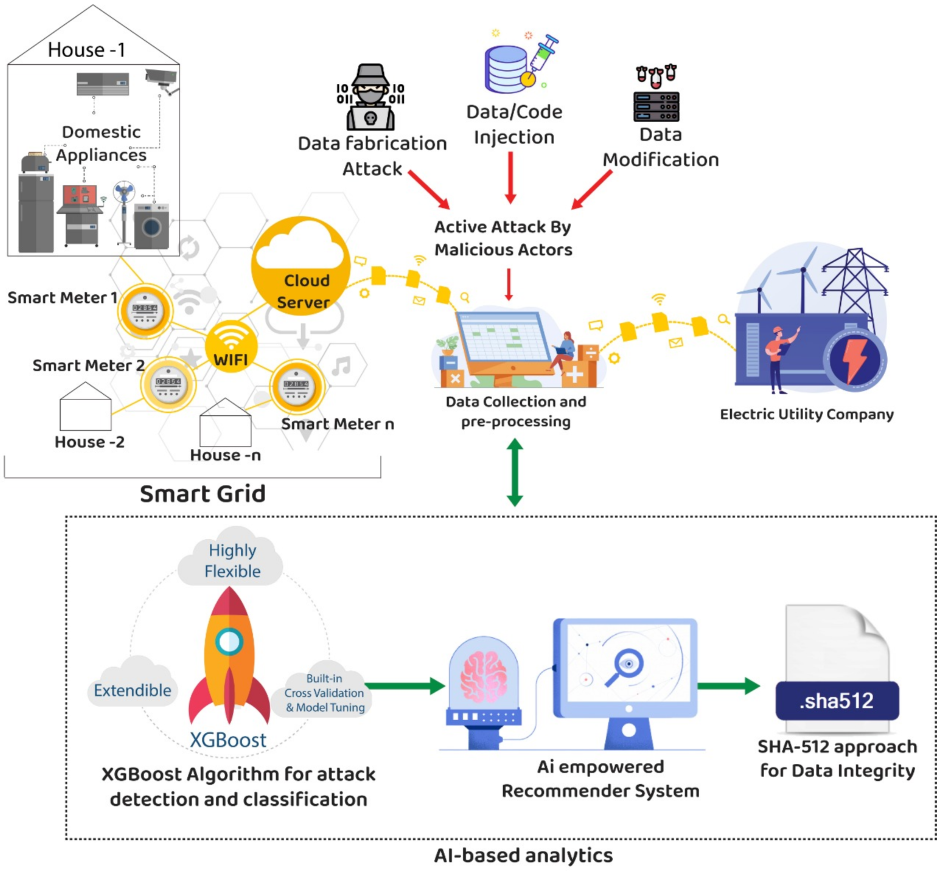


Figure 2. AI-ADP System Model.

3.2. Problem Formulation

In SG, the IoT-based SM generates a lot of data $\mathfrak{ED}$. The $\mathfrak{ED}$ is then transferred from SM to *EUC* and vice versa.

Let, $\{\mathfrak{ED}_1, \mathfrak{ED}_2, \mathfrak{ED}_3, \ldots, \mathfrak{ED}_n\} \in \mathfrak{ED}$ be the data transferred from n end-customer to the *EUC*. Here, the proposed scheme comprises XGBoost for attack detection by using classification and regression tree (CART) to learn continuous and discrete features. Here, CART facilitates to distributed and parallel implementation to detect prominent attacks. For $\mathfrak{ED}_{\mathfrak{n}}$ with f features, XGBoost incorporates c CART base learners to perform an ultimate classification decision. The objective function can be defined as follows:

$$\mathbb{O}_{\mathbb{XG}} = \sum_i \Lambda(\overline{\gamma}_i, \gamma_i) - \sum_j \Delta(f_j). \tag{1}$$

Here, $\Lambda(\overline{\gamma}_i, \gamma_i)$ is the loss function that shows the difference of predicted values ($\overline{\gamma}_i$) and the target value (γ_i) with regularization term $\Delta(f_j)$. Furthermore, the proposed AI-ADP scheme uses SHA-512 to prevent any kind of attacks during data communication. Consequently, the objective to secure the communication can be represented as follows:

$$\mathbb{O}_{\mathbb{SHA}} = SHA(\mathfrak{ED}) + min(Cycles), \tag{2}$$

where $SHA(\mathfrak{ED})$ shows the securing $\mathfrak{ED}$ by using cryptography with minimum cycles used per bytes, which is represented as $min(Cycles)$. Hence, the objective function of the proposed AI-ADP scheme comprises attack detection and maximizes the security of $\mathfrak{ED}$ by using cryptography with minimum cycles used per bytes, which is defined by using Equations (1) and (2) as follows:

$$\begin{aligned} \mathbb{O} &= \mathbb{O}_{\mathbb{XG}} + max(\mathbb{O}_{\mathbb{SHA}}) \\ &\leftarrow \sum_i l(\overline{\gamma}_i, \gamma_i) - \sum_j \Omega(f_j) \\ &+ max(SHA(\mathfrak{ED}) + min(Cycles)). \end{aligned} \tag{3}$$

The defined objective for attack detection and prevention is subject to the following constraints:

$$n, f, i, j, c > 0 \tag{4}$$

$$\{\mathfrak{ED}, EUC\} \neq \phi, \tag{5}$$

where a number of the consumer must be at least one and features should be more than zero. Then, at least one EUC must be there to detect an attack and employ the proposed scheme.

4. AI-ADP: The Proposed Scheme

Figure 3 illustrates the workflow of the proposed AI-ADP scheme, which could be divided into two layers (i) data preparation and (ii) AI-based modeling and securing the system. The first layer comprises the selection of features, then sorting of data elements, normalization, and transformation. Once $\mathfrak{ED}$ is preprocessed, then it is transferred to the second layer for attack detection in $\mathfrak{ED}$ by using XGBoost with prevention of it by employing SHA-512.

4.1. Data Preparation

$\mathfrak{ED}$ consists of noise, missing values, inappropriate values, Not a Number (NaN), and other issues during its collection from SM in the SG system. As a result, once we receive the raw $\mathfrak{ED}$ from SM, it requires pre-processing before using it. For data preprocessing, we employ a linear interpolation (LI) mechanism. LI is a simple technique for estimating unknown values in the linearly spaced values between the two closest defined points. The knowledge of two $\mathfrak{ED}$ points and their constant rate of change is required here [37]. Next, the preprocessed data is transformed from a 2D array to a 3D one for attack detection.

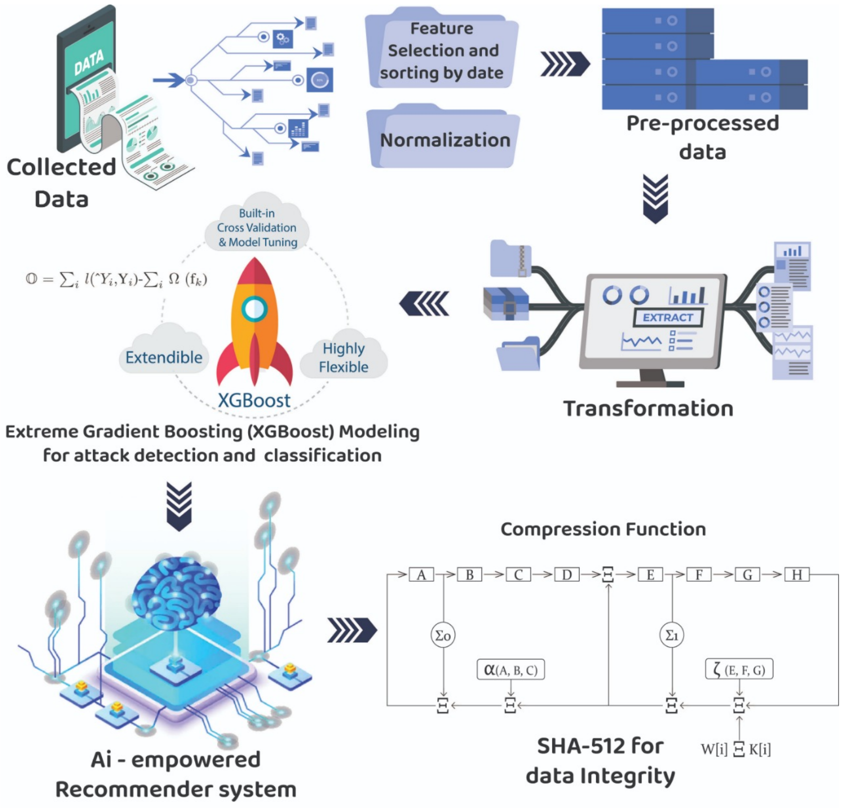

DATA
Collected Data
Feature Selection and sorting by date
Normalization
Pre-processed data
Transformation
EXTRACT
Built-in Cross Validation & Model Tuning
$\mathbb{O} = \sum_i l(\hat{}Y_i,Y_i)-\sum_i \Omega\ (f_k)$
Extendible
Highly Flexible
XGBoost
Extreme Gradient Boosting (XGBoost) Modeling for attack detection and classification
Ai - empowered Recommender system
Compression Function
A
B
C
D
E
F
G
H
Σo
Σ1
α(A, B, C)
ζ (E, F, G)
W[i]
K[i]
SHA-512 for data Integrity

Figure 3. AI-ADP: The workflow.

4.2. AI-Based Modeling and Securing the System

For classification and detection, the AI-ADP scheme employs an AI-based XGBoost model for attack classification and detection. XGBoost is a learning technique to boost the $\mathfrak{ED}$ balance and afterward improve the performance of AI-ADP for a class of normal, attack events, or faulted.

Some of the features of XGBoost that are included in the proposed AI-ADP scheme to make it more robust, for example, tree pruning, flexibility, sparsity-aware split finding, etc. Here, XGBoost pruning is a strategy for shrinking regression trees by removing nodes that don’t contribute to better leaf categorization. In the AI-ADP scheme, XGBoost builds nodes (also known as splits) up to the specified max depth and then begins pruning from the backward direction until the loss is below a threshold (*thres*). Then, in sparsity-aware split finding, a default direction is applied to find the sparsity pattern. The AI-ADP allocates the default direction and selects the optimum imputation value to minimize training loss.

In AI-ADP scheme, the XGBoost-based attack detection model is trained by using an addictive mechanism that means optimization is performed in every iteration in spite of the after-model training. The additive training can be represented as follows [38]:

$$\overline{\gamma}_i^{(t)} = \sum_{j=1}^{t}(F_j(\eta_i)) = \overline{\gamma}_i^{(t-1)} + F_t(\eta_i), \tag{6}$$

where, at the tth iteration, the optimized tree structure, i.e., F_j, minimizes the objective. Here, η_i represents the features information of the ith sample from the dataset, $\gamma_i^{(t)}$ shows the actual data value of the ith sample in tth iteration, and $\overline{\gamma}_i^{(t)}$ depicts the predicted data value of the ith sample in tth iteration.

The Taylor function is employed in objective $\mathbb{O}_{\mathbb{XG}}$ that is denoted as follows [38]:

$$\mathbb{O}^{(\approx)} = \sum_{i=1}^{n} \Lambda(\overline{\gamma}_i^{(t-1)} + F_t(\eta_i), \gamma_i) + \Delta(F_t) + C$$

$$\simeq \sum_{i=1}^{n}[\Lambda(\overline{\gamma}_i^{(t-1)}, \gamma_i) + A_i F_t(\eta_i) + \frac{1}{2}\Gamma_i F_t^2(\eta_i)] + \Delta(F_t) + C, \tag{7}$$

where, $A_i = 2(\overline{\gamma}^{(t-1)} - \gamma_i), \Gamma_i = 2$ and C is a constant, which can be removed to simplify objective function for tth iteration, defined as follows:

$$\mathbb{O}^{(\approx)} = \sum_{i=1}^{n}[A_i F_t(\eta_i) + \frac{1}{2}\Gamma_i F_t^2(\eta_i)] + \Delta(F_t). \tag{8}$$

The objective function can be encouraged to minimize as follows:

$$\mathbb{O}^{(\approx)} = \sum_{i=1}^{n}[A_i \varrho_{t(\eta_i)} + \frac{1}{2}\Gamma_i \varrho^2_{t(\eta_i)}] + \delta T + \frac{1}{2}\lambda \sum_{j=1}^{T} \varrho_j^2. \tag{9}$$

Here, $\Delta(F\_t)$ is calculated as follows:

$$\Delta(F_t) = \delta T + \frac{1}{2}\lambda \sum_{j=1}^{T} \varrho_j^2, \tag{10}$$

where, δ represents the penalty coefficient of the score of the leaf node, $\varrho_{t(\eta_i)}$ shows result of the the decision tree, T represents the number of leaf nodes of the tree, and δ represents the contraction coefficient of T.

Furthermore, the below scoring function is used for optimal decision making (i.e., optimal splitting point) in the Equation (3) by using a greedy algorithm that tries to join a split node to leaf node in every iteration:

$$Gain = \frac{1}{2}[\frac{\kappa_L^2}{\varphi_L + \lambda} + \frac{\kappa_R^2}{\varphi_R + \lambda} - \frac{(\kappa_L + \kappa_R)^2}{\varphi_L + \varphi_R + \lambda}] - \Delta. \tag{11}$$

Then, Δ is the coefficient of the regularization term, and κ represents the summation of the first derivatives after the splitting of the node, and φ depicts the summation of the second derivatives after node splitting. Next, R and L show the right and left nodes, respectively.

After the attack identification and classification, the victimized SM must be secured by using the SHA-512 method. Algorithm 1 presents the steps for the proposed AI-ADP scheme for XGBoost-based attack detection. Next, the XGBoost method performs binary classification of data, i.e., attacked and nonattacked data. The proposed AI-ADP scheme discussed several attacks like SQL injection and data-modification attacks through invaded SM in residential houses. Furthermore, to prevent the attack, there are numerous cryptographic algorithms for data, but SHA-512 is one of the safest and best for data integrity. Therefore, Algorithm 2 shows the prevention mechanism that shows the feasibility of SHA-512 for data integrity. It would be expensive and computationally time-consuming to implement SHA-512 for all houses. Therefore, this paper suggests the adoption of SHA-512 only in the attacked areas or households with the aid of a recommender system. Additionally, it is not practical to implement SHA-512 for files with fewer than 60 characters, so the recommender system also considers the file's character count, which should be more than the minimum number of characters (NC) defined by the system. Here, NC is a system-dependent parameter.

Figure 4 shows the message hashing process in SHA-512. On 64-bit processors, SHA-512 is faster than other algorithms like SHA-256 because it has 37.5% fewer cycles per byte.

In the AI-ADP scheme, $\mathfrak{ED}$ data is first hashed at the end-consumer's end, i.e., sender, and then hash verification is done at another end-consumer's end, i.e., the receiver's end by using SHA-512. The hash value (i.e., a fixed-size output of enciphered text) generated by using the SHA-512 method is almost impossible for MA to breach/regenerate. Here, hash value generation requires more time and resources by MAs compared to the other existing cryptography mechanism. This shows the potential of SHA-512 with its high-security feature.

Algorithm 1 XGBoost-based attack detection.

```
Input: $\mathfrak{ED}_{\mathfrak{train}}$ , $\mathfrak{ED}_{\mathfrak{test}}$
Output: Hashed_file
$\mathfrak{TR}\_X$ = $\mathfrak{ED}_{\mathfrak{train}}$_seperate(independent_variable, target)
$\mathfrak{TR}\_Y$ = $\mathfrak{ED}_{\mathfrak{train}}$(target)
$\mathfrak{TE}\_X$ = $\mathfrak{ED}_{\mathfrak{test}}$_seperate(independent_variable, target)
$\mathfrak{TE}\_Y$ = $\mathfrak{ED}_{\mathfrak{test}}$(target)
XGBoost_fit($\mathfrak{TR}\_X$, $\mathfrak{TR}\_Y$)
p_train = XGBoost_predict($\mathfrak{TR}\_X$)
p_test = XGBoost_predict($\mathfrak{TE}\_X$)
a_train = XGBoost_score($\mathfrak{TR}\_Y$, p_train)
a_test = XGBoost_score($\mathfrak{TE}\_Y$, p_test) // Calling hashing function
for $i \leftarrow 1$ to N do
    count = 0
    for $j \leftarrow 1$ to m do
        if $\mathfrak{a}[i,j] = 1$ then //1 indicated that the data is attacked
            count = count + 1
        end if
    end for
    $\mathbb{B}[i]$ = count
end for
// Securing SG energy data in equivalent hexadecimal value using Algorithm 2
```

Algorithm 2 Recommend to implement SHA-512 for attack prevention.

```
procedure RECOMMENDER(int $\mathbb{B}[i]$, int $\mathbb{L}$)
    for $i \leftarrow 1$ to N do
        if ($\mathbb{B}[i] \geq thres$ & $\mathbb{L} \geq NC$) then
            Recommend $SHA\_512()$
            $\mathbb{M} = Input\_message(originalmessage)$ // Import a energy data file.
            $\mathbb{F}\_message = append\_paddingBits(\mathbb{M})$
            $\mathbb{HS} = hash\_buffer(\mathbb{F}\_message)$ //dividing into data blocks.
            for $i \leftarrow 0$ to $N$ do
                h(i) = $hash\_SHA$-512($\mathbb{HS}[i]$) + h(i−1)
            end for
            $\mathbb{X}_1$ = h(N) //final hash value
            Receiver $\leftarrow (\mathbb{M}, \mathbb{X}_1)$
            $\mathbb{X}_2 = hash\_SHA\text{-}512(\mathbb{M})$
            if $\mathbb{X}_1 = \mathbb{X}_2$ then
                Data ($\mathbb{M}$) is secured and integrity is maintained.
            else
                Data has been attacked through SQL injection, DoS attack, etc.
            end if
        else
            Recommend $Advance\_SHA\_algo()$
        end if
    end for
end procedure
```

SHA-512 embrace preprocessing and then perform hash computation. Initially, eight variables are initialized for fixed constants; then, the input message gets padded and divided into blocks of 1024-bit size. Here, actual hash computation is done by using a message schedule of 16 message blocks. Figure 4 illustrate the message schedule of different message blocks, i.e., M[0], M[1], . . . , M[15] of 64 bits each [39]. The message is passed through the message schedule to augment them to 80 words ω_q, where $0 \leq q \leq 79$. Next, message variables are updated with the help of a compression function. The various functions, such as $\sigma[0/1, i], \omega(q - k)$, etc., uses SHA-512 to secure the $\mathfrak{CD}$ with maximum security, discussed as follows:

$$\sigma[0, q] = RR(\omega_{q-15}, 1) \bigoplus RR(\omega_{q-15}, 8) \bigoplus RR(\omega_{q-15}, 7) \tag{12}$$

$$\sigma[1, q] = RR(\omega_{q-2}, 19) \bigoplus RR(\omega_{q-2}, 61) \bigoplus RR(\omega_{q-2}, 6) \tag{13}$$

$$\omega[q] = \omega_{q-16} \,\Xi\, \sigma[0, q] \; \omega_{q-7} \,\Xi\, \sigma[1, q] \tag{14}$$

$$\Sigma[0, q] = RR(A_q, 28) \bigoplus RR(A_q, 34) \bigoplus RR(A_q, 39) \tag{15}$$

$$\alpha[i] = (A_q \wedge B_q) \bigoplus (A_q \wedge C_q) \bigoplus (B_q \wedge C_q) \tag{16}$$

$$T[2, q] = \Sigma[0, q] \,\Xi\, \alpha[q] \tag{17}$$

$$\Sigma[1, q] = RR(E_q, 14) \bigoplus RR(E_q, 18) \bigoplus RR(E_q, 41) \tag{18}$$

$$\zeta[q] = (E_q \wedge F_q) \bigoplus (E_q \wedge G_q), \tag{19}$$

where $sigma[0/1, q]$ is the sigmoid function, RR shows the right rotation of the bits based on the $sigma[0/1, q]$ function, and Ξ is the compression function used in the hashing technique. Each cycle of the compression function consists of seven modular 64-bit word additions and four substantially more expensive operations: α (Majority) and $\zeta[q]$ (Choice) are made up of a series of bitwise operations (mostly logical AND and XOR), and the $sigma[0/1, q]$ perform XORs and bit-position rotations on a 64-bit word.

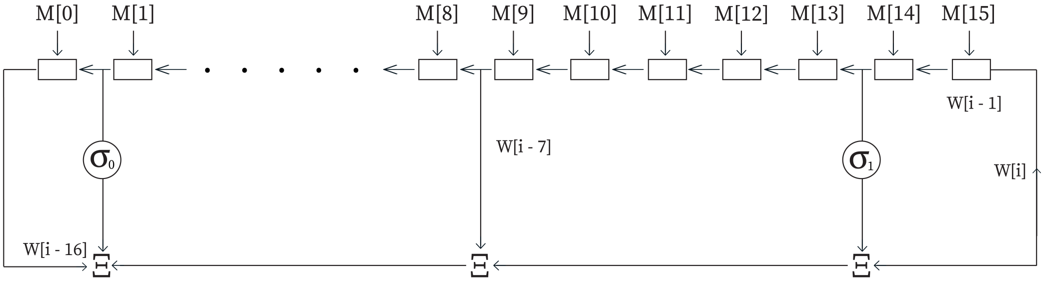


Figure 4. The SHA-512 message schedule.

5. Performance Evaluation

In this section, we perform the simulations and evaluate the performance of AI-ADP scheme with respect to the attack-detection accuracy and cycles used per byte. The elaboration of the same is discussed in-depth for attack-detection accuracy and securing the SG from attacks by using SHA-512.

5.1. Dataset Descriptions

The performance analysis of AI-ADP includes energy-consumption data on an hourly basis. The smart-home data is referred from a standard benchmarked openEI dataset (comprising real-time energy data) [40] that contains energy-consumption values in kilowatts per hour (kWh) of the residential sector. This dataset provides useful insights to technology enthusiasts, energy traders, researchers, and policymakers. The dataset comprises hourly residential houses' energy consumption data by using various home appliances such as lighting, basic facilities, AC, and miscellaneous (Misc) for Alaska-based houses. The dataset does not comprise attacked values, so we generated the attacked dataset from the openEI datasets and used oversampling. To generate the attacked dataset, we developed Python APIs and incorporated them with an open-source SQL injection tool, i.e., SQLmap. Next, this tool supports several types of SQL injection attacks, for instance, time-based blind, boolean-based blind, UNION query-based, error-based, out-of-band, and stacked queries. Furthermore, it comes with powerful features like editing, uploading, and downloading files in a database and facilitates a detection engine that can easily detect SQL injection vulnerabilities. Then, the K-Means-based synthetic minority oversampling technique (SMOTE) is used. This mechanism circumvents the noise generation and effectively handles imbalances within the classes of attacked and nonattacked data. Then, the dataset is divided into a 75:25 ratio to train the XGBoost model and test the model accuracy on the test dataset.

5.2. Experimental Setup and Tools

The AI-ADP scheme is evaluated in the windows OS with the following configurations:

- Intel(R) Core(TM) CPU (Intel Core i7 @ 2.6 GHz);
- 16 GB memory;
- 250 GB SSD; and
- 1 Gbit/s network.

The evaluation of the AI-ADP is done in the Windows-10 system by using Python computer programming language. Then, all the simulation parameters are set initially by using Table 3.

Figure 5 illustrates the comparison between predicted $\mathfrak{ED}$ and attacked $\mathfrak{ED}$ for a particular end-customer for the 60 h. It depicts from the graph that attacked $\mathfrak{ED}$ is quite far from the actual $\mathfrak{ED}$ values. Detection of attacks in the SG system is a crucial component; hence, correct detection is essential. Figure 6 shows the confusion matrix for the proposed AI-ADP scheme that illustrates the co-relation of the correct detection of attack with the nonattack $\mathfrak{ED}$. Furthermore, the AI-ADP scheme obtained 99.12% accuracy, which shows its effectiveness compared to the existing baseline works that gained less accuracy [41,42].

In the beginning, the sender needs to generate a hash value for the data file. The data file can be in various formats like comma separated values (CVS), portable document format (PDF), etc. After importing a data file, the proposed scheme scans and generates the hash string in blocks of 4 K and generates the final hash value of length 128 in hexadecimal form, as shown in the Figure 7.

Furthermore, at the receiver's end, the receiver receives the data file and hash value of the file. The hash value of the received file is calculated by the receiver using the same procedure that the sender used to hash the file employing SHA-512. After generating the final hash value, it is compared to the hash value shared by the sender with the receiver.

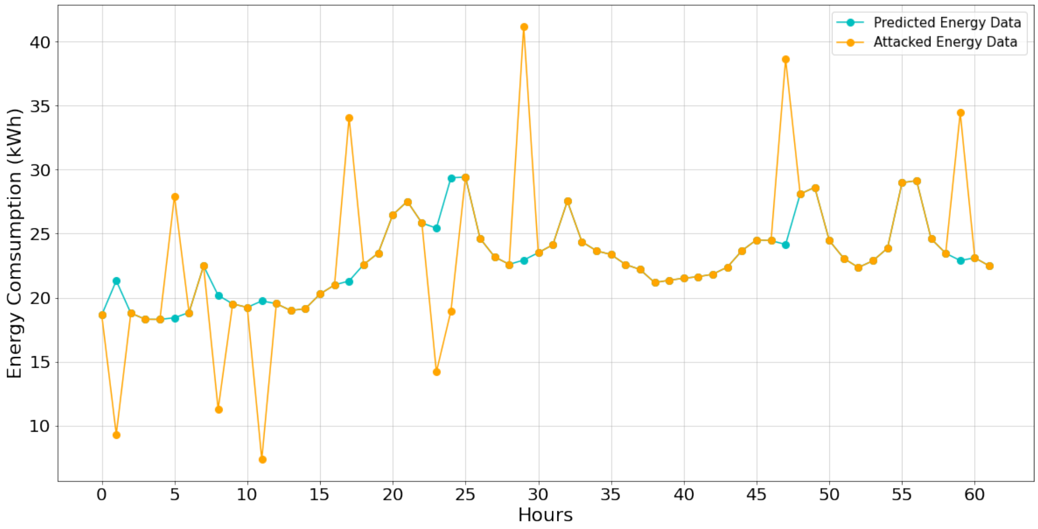


Figure 5. Attack detection in the proposed AI-ADP scheme.

Table 3. Simulation setting.

Parameters	Configuration
Size of hash value	512
Message block size	1024
Internal state size	512
Maximum message size	$2^{128} - 1$
Complexity of the best attack	2^{256}
Word size	64
Number of words	8
Number of digest rounds	80
Constants Kt number	80

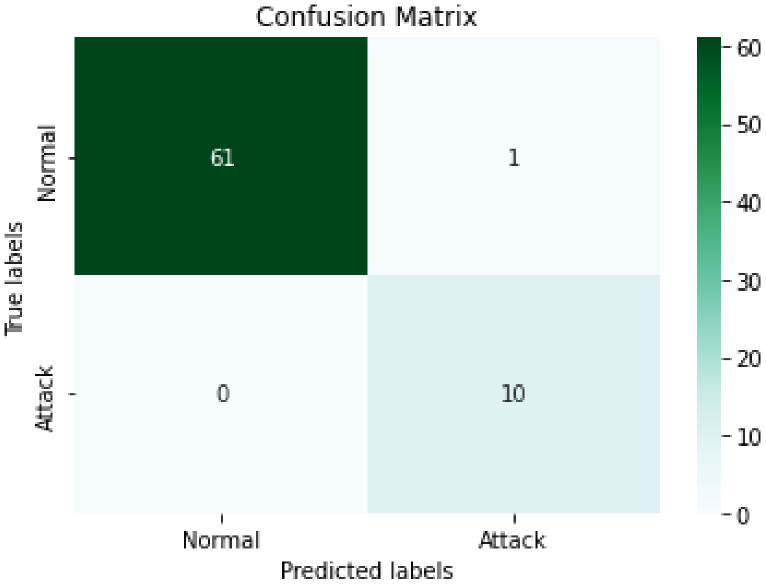
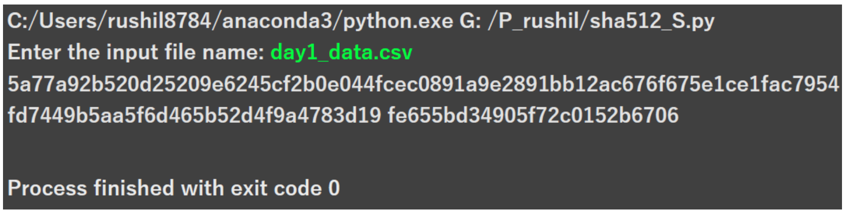


Figure 6. Confusion matrix for the proposed AI-ADP scheme.



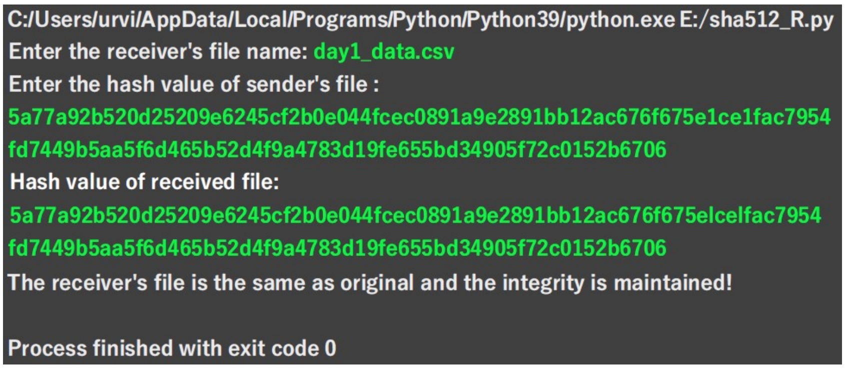
```
C:/Users/rushil8784/anaconda3/python.exe G: /P_rushil/sha512_S.py
Enter the input file name: day1_data.csv
5a77a92b520d25209e6245cf2b0e044fcec0891a9e2891bb12ac676f675e1ce1fac7954
fd7449b5aa5f6d465b52d4f9a4783d19 fe655bd34905f72c0152b6706


Process finished with exit code 0
```

Figure 7. Sender’s data hash value generation.

Figure 8 shows data integrity and security preserved in the proposed scheme if hash values are the same at the receiver end. Next, when hash values are different, data has been attacked and data integrity has been violated, that is depicted in Figure 9.

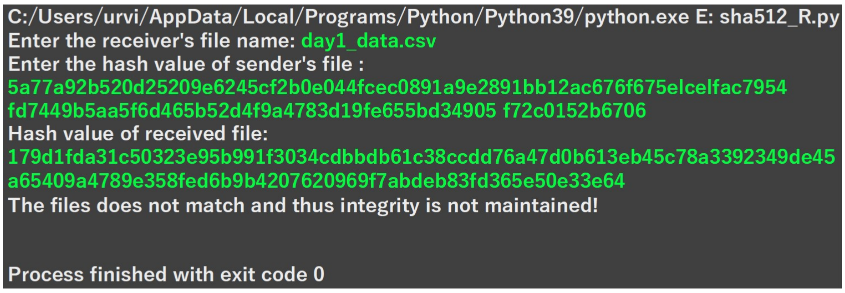
Figure 10 presents a comparative analysis between the AI-ADP scheme and the existing approaches (Existing Approach-1 [43]) and (Existing Approach-2 [44]). For the purpose of evaluating the AI-ADP scheme, four parameters are taken: (A) Approach Update {Compact C code}, (B) Approach Update {OpenSSL unrolled asm code}, (C) Approach on a 1024-byte message {Compact C code}, and (D) Approach on a 1024-byte message {OpenSSL unrolled asm code}. Here, Figure 10(i) clearly shows that AI-ADP outperformed compare to preexisting methods with respect to fewer cycles used per byte in the proposed scheme for four test cases. Similarly, Figure 10(ii) displays the total number of cycles required for text/message and hashing for four parameters A, B, C, and D. Here, total cycles used range from 0 to 35,000 and due to implementation constraints, total cycles used values are represented in numbers, not in a string (e.g., 20,000 is shown in place of 20,000). It depicts from the graph that fairly fewer cycles are used as a total in the proposed AI-ADP scheme compared to Existing Approach-1 [43] and Existing Approach-2 [44].

Attack detection becomes an important direction in future work by exploring it further by using neural-network-based models such as artificial neural networks, recurrent neural networks, etc. Then, one of the challenges with SHA-512 is that it necessitates a table of eighty 64-bit constants, i.e., a 640-byte lookup table. The cost of storing data in a lookup table is quite costly in some systems. Next, we will explore an advanced cryptography mechanism to secure the data communication in SG system with real-time accessibility.



```
C:/Users/urvi/AppData/Local/Programs/Python/Python39/python.exe E:/sha512_R.py
Enter the receiver's file name: day1_data.csv
Enter the hash value of sender's file :
5a77a92b520d25209e6245cf2b0e044fcec0891a9e2891bb12ac676f675e1ce1fac7954
fd7449b5aa5f6d465b52d4f9a4783d19fe655bd34905f72c0152b6706
Hash value of received file:
5a77a92b520d25209e6245cf2b0e044fcec0891a9e2891bb12ac676f675elcelfac7954
fd7449b5aa5f6d465b52d4f9a4783d19fe655bd34905f72c0152b6706
The receiver's file is the same as original and the integrity is maintained!

Process finished with exit code 0
```

Figure 8. Receiver received the original data.



```
C:/Users/urvi/AppData/Local/Programs/Python/Python39/python.exe E: sha512_R.py
Enter the receiver's file name: day1_data.csv
Enter the hash value of sender's file :
5a77a92b520d25209e6245cf2b0e044fcec0891a9e2891bb12ac676f675elcelfac7954
fd7449b5aa5f6d465b52d4f9a4783d19fe655bd34905 f72c0152b6706
Hash value of received file:
179d1fda31c50323e95b991f3034cdbbdb61c38ccdd76a47d0b613eb45c78a3392349de45
a65409a4789e358fed6b9b4207620969f7abdeb83fd365e50e33e64
The files does not match and thus integrity is not maintained!


Process finished with exit code 0
```

Figure 9. Receiver received the attacked data.

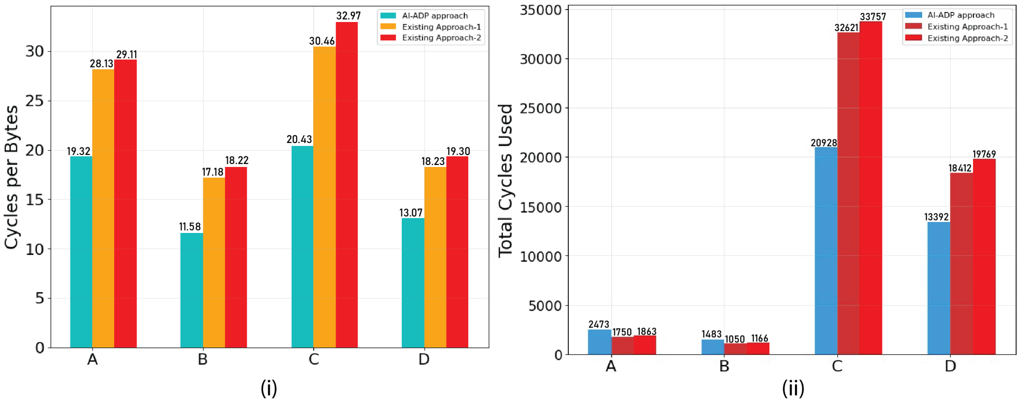


Figure 10. A comparative analysis of the AI-ADP scheme with existing approaches: (**i**) Cycles used per byte and (**ii**) total cycles used.

6. Discussion

The proposed work presents secure communication in the SG system by adopting indispensable features of AI and cryptography. The scientific community has already delivered an outstanding solution to protect SG data from being vulnerable to several attacks, such as data-modification attacks, SQL injection attacks, etc. However, they lacked in offering protection to critical data exchange between end-to-end systems. Moreover, the solutions, for instance, AI and cryptography, are solely encountered security attacks. Additionally, many researchers use the traditional Internet, which always has a high risk of being attacked. Consequently, there is a rigorous requirement for a mechanism that constantly supervises the bidirectional communication in an SG environment. The proposed AI-ADP employs an XGBoost model to train on different types of attacks like data modification and SQL injection attacks to bifurcate the nonmalicious and malicious data in the SG system. Further, to handle this issue, the SG system needs to ensure that data sent from the sender is reaching in the same form at the receiver end. Therefore, an algorithm is designed for data-integrity check and attack prevention in the SG system with the help of SHA-512 cryptography. This enables one to hash the data and lowers the risk of processing malicious data in subsystems of an SG. Therefore, we evaluate the performance of the proposed AI-ADP in the residential environment by using residential datasets, proving that the proposed AI-ADP is far better than the existing baseline works.

7. Conclusions

SG is one of the critical infrastructures where even a single attack can have a possibly disastrous impact. Integrating SG with advanced technologies could increase the risk of attacks and cyberthreats in a cyber-physical system of SG. Consequently, this paper presents a secure AI-based attack-detection and prevention methodology, i.e., the AI-ADP scheme. Here, in the proposed scheme, an attack is identified by using a prominent XGBoost mechanism, and then the SM's data is safeguarded by using an SHA-512 cryptographic technique through a recommender system. The data is effectively hashed after using the SHA-512 method, making it almost impossible for attackers to breach/regenerate it. The proposed scheme is evaluated compared to other pre-existing schemes based on the parameters like cycles used per bytes, total cycles used, and attack-detection accuracy. As a result, the goal of this paper is to secure the SG to a higher level.

In the near future, attacks detection will be extended further with detection accuracy by using recurrent neural networks, and advanced cryptography mechanisms will be explored to handle data-storage cost issues in the lookup table of SHA-512 in the SG environment.

Author Contributions: Conceptualization: A.K., R.K.P. and U.C.S.; writing—original draft preparation: S.T., M.S.R. and A.T.; methodology: A.S., S.T., U.C.S. and A.T.; writing—review and editing: A.S., R.K.P., A.K. and A.T.; investigation: M.S.R., A.K., S.T. and U.C.S.; supervision: A.K., A.T. and S.T. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to extend their gratitude to King Saud University (Riyadh, Saudi Arabia) for funding this research through Researchers Supporting Project number (RSP-2021/260).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to extend their gratitude to King Saud University (Riyadh, Saudi Arabia) for funding this research through Researchers Supporting Project number (RSP-2021/260). In addition, this paper was also supported by Subprogram 1.1. Institutional performance-Projects to finance excellence in RDI, Contract No. 19PFE/30.12.2021 and a grant of the National Center for Hydrogen and Fuel Cells (CNHPC)—Installations and Special Objectives of National Interest (IOSIN). This paper was partially supported by UEFISCDI Romania and MCI through projects AISTOR, FinSESco, CREATE, I-DELTA, DEFRAUDIFY, Hydro3D, EREMI, SMARDY, STACK, ENTA, PREVENTION, RECICLARM, UPSIM, SmartDelta, USWA and by European Union's Horizon 2020 research and innovation program under grant agreements and No. 777996 (SealedGRID), No. 872172 (TESTBED2), No. 883522 (S4ALLCITIES).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hossain, E.; Khan, I.; Un-Noor, F.; Sikander, S.S.; Sunny, M.S.H. Application of Big Data and Machine Learning in Smart Grid, and Associated Security Concerns: A Review. *IEEE Access* **2019**, *7*, 13960–13988. [CrossRef]
2. Tan, S.; De, D.; Song, W.Z.; Yang, J.; Das, S.K. Survey of Security Advances in Smart Grid: A Data Driven Approach. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 397–422. [CrossRef]
3. Kumari, A.; Tanwar, S. A Data Analytics Scheme for Security-aware Demand Response Management in Smart Grid System. In Proceedings of the 2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Prayagraj, India, 27–29 November 2020; pp. 1–6. [CrossRef]
4. Ericsson, G.N. Cyber Security and Power System Communication—Essential Parts of a Smart Grid Infrastructure. *IEEE Trans. Power Deliv.* **2010**, *25*, 1501–1507. [CrossRef]
5. Kumari, A.; Patel, M.M.; Shukla, A.; Tanwar, S.; Kumar, N.; Rodrigues, J.J.P.C. ArMor: A Data Analytics Scheme to identify malicious behaviors on Blockchain-based Smart Grid System. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 8–10 December 2020; pp. 1–6. [CrossRef]
6. Hu, C.; Yan, J.; Wang, C. Advanced Cyber-Physical Attack Classification with eXtreme Gradient Boosting for Smart Transmission Grids. In Proceedings of the 2019 IEEE Power & Energy Society General Meeting (PESGM), Atlanta, GA, USA, 4–8 August 2019; pp. 1–5.
7. Kumari, A.; Tanwar, S. A Reinforcement-Learning-Based Secure Demand Response Scheme for Smart Grid System. *IEEE Internet Things J.* **2022**, *9*, 2180–2191. [CrossRef]
8. Oliveira, N.; Praça, I.; Maia, E.; Sousa, O. Intelligent Cyber Attack Detection and Classification for Network-Based Intrusion Detection Systems. *Appl. Sci.* **2021**, *11*, 1674. [CrossRef]
9. Farrukh, Y.A.; Khan, I.; Ahmad, Z.; Elavarasan, R.M. A Sequential Supervised Machine Learning Approach for Cyber Attack Detection in a Smart Grid System. *arXiv* **2021**, arXiv:2108.00476.
10. Khan, A.A.; Kumar, V.; Ahmad, M. An Elliptic Curve Cryptography Based Mutual Authentication Scheme for Smart Grid Communications Using Biometric Approach. *J. King Saud Univ.-Comput. Inf. Sci.* **2019**, *34*, 698–705. doi: 10.1016/j.jksuci.2019.04.013. [CrossRef]
11. Manandhar, K.; Cao, X.; Hu, F.; Liu, Y. Detection of Faults and Attacks Including False Data Injection Attack in Smart Grid Using Kalman Filter. *IEEE Trans. Control Netw. Syst.* **2014**, *1*, 370–379. [CrossRef]
12. Kurt, M.N.; Ogundijo, O.; Li, C.; Wang, X. Online Cyber-Attack Detection in Smart Grid: A Reinforcement Learning Approach. *IEEE Trans. Smart Grid* **2019**, *10*, 5174–5185. [CrossRef]
13. Kumari, A.; Tanwar, S. A secure data analytics scheme for multimedia communication in a decentralized smart grid. *Multimed. Tools Appl.* **2021**, 1–26. [CrossRef]
14. Mitali, V.K.; Sharma, A. A Survey on Various Cryptography Techniques. *Int. J. Emerg. Trends Technol. Comput. Sci.* **2014**, *3*, 307–312.
15. Gupta, B.; Agrawal, D.P.; Yamaguchi, S. *Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security*; IGI Global: Hershey, PA, USA, 2016; pp. 1–589.
16. Sadhukhan, D.; Ray, S.; Obaidat, M.S.; Dasgupta, M. A Secure and Privacy Preserving Lightweight Authentication Scheme for Smart-Grid Communication using Elliptic Curve Cryptography. *J. Syst. Archit.* **2021**, *114*, 101938. [CrossRef]
17. Rezaimehr, F.; Dadkhah, C. A survey of Attack Detection Approaches in Collaborative Filtering Recommender Systems. *Artif. Intell. Rev.* **2021**, *54*, 2011–2066. [CrossRef]
18. Rubio, J.E.; Alcaraz, C.; Lopez, J. Recommender System for Privacy-Preserving Solutions in Smart Metering. *Pervasive Mob. Comput.* **2017**, *41*, 205–218. [CrossRef]
19. Patel, R.K.; Kumari, A.; Tanwar, S.; Hong, W.C.; Sharma, R. AI-Empowered Recommender System for Renewable Energy Harvesting in Smart Grid System. *IEEE Access* **2022**, *10*, 24316–24326. [CrossRef]
20. Yan, Y.; Qian, Y.; Sharif, H.; Tipper, D. A Survey on Cyber Security for Smart Grid Communications. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 998–1010. [CrossRef]
21. Liu, J.; Xiao, Y.; Li, S.; Liang, W.; Chen, C.L.P. Cyber Security and Privacy Issues in Smart Grids. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 981–997. [CrossRef]
22. Wang, S.; Bi, S.; Zhang, Y.J.A.; Huang, J. Electrical Vehicle Charging Station Profit Maximization: Admission, Pricing, and Online Scheduling. *IEEE Trans. Sustain. Energy* **2018**, *9*, 1722–1731. [CrossRef]
23. Inayat, U.; Zia, M.F.; Mahmood, S.; Khalid, H.M.; Benbouzid, M. Learning-Based Methods for Cyber Attacks Detection in IoT Systems: A Survey on Methods, Analysis, and Future Prospects. *Electronics* **2022**, *11*, 1502. [CrossRef]
24. Dobraunig, C.; Eichlseder, M.; Mendel, F. Analysis of SHA-512/224 and SHA-512/256. In *Advances in Cryptology—ASIACRYPT 2015*; Iwata, T., Cheon, J.H., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; pp. 612–630.
25. Bansal, A.; Kaur, S. Extreme Gradient Boosting Based Tuning for Classification in Intrusion Detection Systems. In *International Conference on Advances in Computing and Data Sciences*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 372–380.
26. Fan, J.; Wang, X.; Wu, L.; Zhou, H.; Zhang, F.; Yu, X.; Lu, X.; Xiang, Y. Comparison of Support Vector Machine and Extreme Gradient Boosting for Predicting Daily Global Solar Radiation Using Temperature and Precipitation in Humid Subtropical climates: A case Study in China. *Energy Convers. Manag.* **2018**, *164*, 102–111. [CrossRef]

27. Cherif, I.L.; Kortebi, A. On Using eXtreme Gradient Boosting (XGBoost) Machine Learning Algorithm for Home Network Traffic Classification. In Proceedings of the 2019 Wireless Days (WD), Manchester, UK, 24–26 April 2019; pp. 1–6. [CrossRef]
28. Camana Acosta, M.R.; Ahmed, S.; Garcia, C.E.; Koo, I. Extremely Randomized Trees-Based Scheme for Stealthy Cyber-Attack Detection in Smart Grid Networks. *IEEE Access* **2020**, *8*, 19921–19933. [CrossRef]
29. Su, Q.; Li, S.; Gao, Y.; Huang, X.; Li, J. Observer-Based Detection and Reconstruction of Dynamic Load Altering Attack in Smart Grid. *J. Frankl. Inst.* **2021**, *358*, 4013–4027. [CrossRef]
30. Patnaik, B.; Mishra, M.; Bansal, R.C.; Jena, R.K. MODWT-XGBoost Based Smart Energy Solution for Fault Detection and Classification in a Smart Microgrid. *Appl. Energy* **2021**, *285*, 116457. [CrossRef]
31. Khamaiseh, S.Y.; Alsmadi, I.; Al-Alaj, A. Deceiving Machine Learning-Based Saturation Attack Detection Systems in SDN. In Proceedings of the 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Leganes, Spain, 10–12 November 2020; pp. 44–50. [CrossRef]
32. Ashraf, S.; Shawon, M.H.; Khalid, H.M.; Muyeen, S.M. Denial-of-Service Attack on IEC 61850-Based Substation Automation System: A Crucial Cyber Threat towards Smart Substation Pathways. *Sensors* **2021**, *21*, 6415. [CrossRef]
33. Zivkovic, M.; Tair, M.; Venkatachalam, K.; Bacanin, N.; Hubálovskỳ, Š.; Trojovskỳ, P. Novel Hybrid Firefly Algorithm: An Application to Enhance XGBoost Tuning for Intrusion Detection Classification. *PeerJ Comput. Sci.* **2022**, *8*, e956. [CrossRef] [PubMed]
34. Kumari, A.; Tanwar, S. RAKSHAK: Resilient and Scalable Demand Response Management Scheme for Smart Grid Systems. In Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science and Engineering (Confluence), Noida, India, 28–29 January 2021; pp. 309–314. [CrossRef]
35. Chehri, A.; Fofana, I.; Yang, X. Security Risk Modeling in Smart Grid Critical Infrastructures in the Era of Big Data and Artificial Intelligence. *Sustainability* **2021**, *13*, 3196. [CrossRef]
36. Alqahtani, M.; Mathkour, H.; Ben Ismail, M.M. IoT Botnet Attack Detection Based on Optimized Extreme Gradient Boosting and Feature Selection. *Sensors* **2020**, *20*, 6336. [CrossRef]
37. Blu, T.; Thevenaz, P.; Unser, M. Linear Interpolation Revitalized. *IEEE Trans. Image Process.* **2004**, *13*, 710–719. [CrossRef]
38. Li, H.; Cao, Y.; Li, S.; Zhao, J.; Sun, Y. XGBoost Model and Its Application to Personal Credit Evaluation. *IEEE Intell. Syst.* **2020**, *35*, 52–61. [CrossRef]
39. Martino, R.; Cilardo, A. SHA-2 Acceleration Meeting the Needs of Emerging Applications: A Comparative Survey. *IEEE Access* **2020**, *8*, 28415–28436. [CrossRef]
40. OpenEI. Open Energy Information: Smart Meters Data from Houses. Available online: https://openei.org/datasets/files/961/pub (accessed on 5 January 2022).
41. Adhikari, U.; Morris, T.H.; Pan, S. Applying Non-Nested Generalized Exemplars Classification for Cyber-Power Event and Intrusion Detection. *IEEE Trans. Smart Grid* **2016**, *9*, 3928–3941. [CrossRef]
42. Adhikari, U.; Morris, T.H.; Pan, S. Applying Hoeffding Adaptive Trees for Real-Time Cyber-Power Event and Intrusion Classification. *IEEE Trans. Smart Grid* **2017**, *9*, 4049–4060. [CrossRef]
43. Pal, A.; Jolfaei, A.; Kant, K. A Fast Prekeying-Based Integrity Protection for Smart Grid Communications. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5751–5758. [CrossRef]
44. Aghapour, S.; Kaveh, M.; Martín, D.; Mosavi, M.R. An Ultra-Lightweight and Provably Secure Broadcast Authentication Protocol for Smart Grid Communications. *IEEE Access* **2020**, *8*, 125477–125487. [CrossRef]