systems

MDPI

Article

Iot-Based Privacy-Preserving Anomaly Detection Model for Smart Agriculture

Keerthi Kethineni and Pradeepini Gera *

Department of CSE, Koneru Lakashmaiah Education Foundation, Vaddeswaram, Guntur 522302, India
* Correspondence: pradeepini_cse@kluniversity.in

Abstract: Internet of Things (IoT) technology has been incorporated into the majority of people's everyday lives and places of employment due to the quick development in information technology. Modern agricultural techniques increasingly use the well-known and superior approach of managing a farm known as "smart farming". Utilizing a variety of information and agricultural technologies, crops are observed for their general health and productivity. This requires monitoring the condition of field crops and looking at many other indicators. The goal of smart agriculture is to reduce the amount of money spent on agricultural inputs while keeping the quality of the final product constant. The Internet of Things (IoT) has made smart agriculture possible through data collection and storage techniques. For example, modern irrigation systems use effective sensor networks to collect field data for the best plant irrigation. Smart agriculture will become more susceptible to cyber-attacks as its reliance on the IoT ecosystem grows, because IoT networks have a large number of nodes but limited resources, which makes security a difficult issue. Hence, it is crucial to have an intrusion detection system (IDS) that can address such challenges. In this manuscript, an IoT-based privacy-preserving anomaly detection model for smart agriculture has been proposed. The motivation behind this work is twofold. Firstly, ensuring data privacy in IoT-based agriculture is of the utmost importance due to the large volumes of sensitive information collected by IoT devices, including on environmental conditions, crop health, and resource utilization data. Secondly, the timely detection of anomalies in smart agriculture systems is critical to enable proactive interventions, such as preventing crop damage, optimizing resource allocation, and ensuring sustainable farming practices. In this paper, we propose a privacy-encoding-based enhanced deep learning framework for the difficulty of data encryption and intrusion detection. In terms of data encoding, a novel method of a sparse capsule-auto encoder (SCAE) is proposed along with feature selection, feature mapping, and feature normalization. An SCAE is used to convert information into a new encrypted format in order to prevent deduction attacks. An attention-based gated recurrent unit neural network model is proposed to detect the intrusion. An AGRU is an advanced version of a GRU which is enhanced by an attention mechanism. In the results section, the proposed model is compared with existing deep learning models using two public datasets. Parameters such as recall, precision, accuracy, and F1-score are considered. The proposed model has accuracy, recall, precision, and F1-score of 99.9%, 99.7%, 99.9%, and 99.8%, respectively. The proposed method is compared using a variety of machine learning techniques such as the deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN), and long short-term memory (LSTM).

Keywords: intrusion detection; encryption or encoding; malware; smart agriculture; IoT; auto encoder; GRU

Citation: Kethineni, K.; Gera, P. Iot-Based Privacy-Preserving Anomaly Detection Model for Smart Agriculture. *Systems* **2023**, *11*, 304. https://doi.org/10.3390/systems11060304

Academic Editors: Tetiana Hovorushchenko, Ivan Izonin and Hakan Kutucu

Received: 17 April 2023
Revised: 6 June 2023
Accepted: 9 June 2023
Published: 13 June 2023

1. Introduction

Agriculture and farming are regarded as two of the most important and fundamental businesses that benefit humanity greatly and boost a country's GDP [1]. Better farming and agriculture management require technological support and breakthroughs [2]. To enhance the quality and output of agricultural goods at a lower cost and with less human

involvement, a group of tactics known as "smart agriculture" are used [3]. Due to the continuous monitoring of numerous elements in the agricultural sector, smart farming is totally based on Internet of Things technology [4], with numerous qualities including water table and the development of plant and soil properties. Machine learning (ML) algorithms are efficiently used to increase crop yield and lower the risk of crop damage [5]. The use of data analytics to make judgments and suggest the best crops for production in the agricultural sector is also significant.

More than 80% of people on the planet have access to reliable Internet [6]. Spot messaging, immersive two-way video chat, chance mobile contact, social networking, voice-over-Internet protocol (VoIP), mobile phone calls, and e-commerce websites are just a few of the ways in which technology has impacted modern life [7]. IoT is a technology that serves as a future online platform for communication related to technology and the environment. The technology is allegedly safer and more affordable in several industries, including agriculture. With the use of intelligent devices and tools, farms can be tracked and monitored autonomously in the area of agriculture with less human participation As it helps farmers to make judgments about crop management and dynamic farming, smart agriculture demonstrates interest in academia [8]. Crop production may be increased by smart farming, which can also raise a nation's GDP rate.

Crop output is reliant on the existence of vermin and plant disease [9]. Pests used to hide behind leaves during the day to escape the warmness, and by dusk or dark, they were visible on leaves [10]. Therefore, it is too difficult to notice their presence throughout the day. Farmers may cause bugs to spread and multiply uncontrollably if they begin to harm pests [11]. In order to get rid of pests and reduce agricultural damage, a lot of pesticides must be sprayed on crops [12]. Additionally, if vegetables are sprayed with pesticides while they are still growing, the pesticides remain in the plant even after washing. When pests cause severe crop diseases, crops can occasionally become infected with bacteria. To stop the spread of bacteria, the afflicted crops must be thoroughly cleaned. Without effectively addressing the pest problem, this approach will harm the entire agricultural production [13]. As a result, the researchers must consider the problem and should use IoT and ML technologies to improve crop growth and insect detection in agricultural fields [14].

Sensors, connectivity, gateways, location, data analytics, and IoT components are essential components for smart agriculture [15]. By gathering crucial data on numerous agricultural characteristics including fertilizer level, soil moisture, and water level, sensors are utilized to promote accurate farming. Network connectivity, including WIFI, cellular, and ZigBee, is referred to as connectivity. Microcontrollers are referred to as gateways. Arduino, Device Hive, and Raspberry PI are examples of IoT components. Four stages make up the core operating structure of smart agriculture. The sensors first collect data regarding the agricultural elements necessary for a crop's development [16]. To sense various agricultural characteristics, numerous sensors are utilized. The demands and shortfalls in agriculture are then determined by mapping the sensed data with additional criteria] ML algorithms are used to find a solution for the given information. The cycle of smart agriculture is continued in this way.

Security is regarded as a serious issue in IoT-ML-enabled smart agriculture. Location monitoring, data theft, known-key assaults, data pollution, data injection, session hijacking, cyber-agroterrorism, malicious code attacks, and repudiation attacks are just a few of the attacks that can happen in an agricultural setting [17]. Data encryption, hashing, noise mixing, message authentication codes, location concealment, identity-based cryptography, digital signatures, multi-factor authentication, group signatures, fault-tolerance, blind signatures, access control, traceable meta-data, and pseudonyms are the main defenses against these attacks [18]. For secured data transmission, these solutions must be taken into account.

If a fast DDoS attack occurs before the system has been trained or the detection has been performed, it can be difficult to prevent the attack or minimize the damage caused by it. By over-provisioning the network bandwidth capacity, the system can better withstand

DDoS attacks [19]. This can be achieved by upgrading the network infrastructure or using traffic filtering services provided by Internet service providers (ISPs). Configuring firewalls and routers to drop traffic from known malicious IP addresses or blocking traffic to specific ports can help to prevent or mitigate DDoS attacks. Cloud-based protection services can provide protection against DDoS attacks by absorbing or mitigating attack traffic. These services can be implemented quickly and are often more effective than traditional on-premise solutions. By reducing the attack surface of the system, it becomes more difficult for attackers to exploit vulnerabilities. This can be achieved by removing unnecessary services or protocols, patching vulnerabilities, and implementing access controls [20]. By training an attention-based gated recurrent unit neural network (AGRU) model to detect DDoS attacks, the system can quickly identify and respond to attacks. The model can be trained on historical data or using a simulated environment, enabling it to quickly adapt to new attack types. While it is always preferable to have a detection system in place before an attack occurs, the measures outlined above can help mitigate the damage caused by a fast DDoS attack that occurs before the system has been trained or the detection has been performed.

Motivation

An IoT-based privacy-preserving anomaly detection system in smart agriculture is used to improve crop yield and reduce wastage while maintaining data privacy. These data can be used to optimize farming practices and detect any anomalies that could potentially harm the crop yield. Therefore, a privacy-preserving anomaly detection system that can securely process data without revealing sensitive information is essential. Moreover, using attention-based gated recurrent unit (GRU) networks can help to improve the accuracy of anomaly detection. Attention mechanisms enable the model to focus on specific features of the data that are relevant for detecting anomalies. The gated recurrent unit (GRU) network is a type of recurrent neural network that is particularly effective in processing sequential data. Therefore, an IoT-based privacy-preserving anomaly detection system using an attention-based gated recurrent unit network can provide farmers and agricultural organizations with a reliable and secure way to monitor crops and optimize farming practices, while protecting sensitive data. These are the factors that motivated us to conduct this research work.

Contributions:

- The first step in building the model is to collect data from IoT devices deployed in the smart agriculture system. These devices can include sensors for measuring environmental parameters, crop health, soil conditions, weather data, etc. The collected data form the basis for anomaly detection.
- Preserving the privacy of sensitive data is crucial in any IoT-based system. Privacy-preserving techniques such as data anonymization, encryption, or differential privacy can be employed to protect personal and confidential information while ensuring that data retain their utility for anomaly detection. These techniques can help in compliance with privacy regulations and prevent unauthorized access.
- Data are changed into a new encoded format using the sparse capsule-auto encoder approach to make them more resistant to attacks.
- The AGRU neural network model is utilized for detecting anomalies in the privacy-preserved data. An AGRU extends the capabilities of the traditional GRU model by incorporating an attention mechanism. The attention process allows the model to focus on important features or time steps, effectively capturing temporal dependencies and identifying abnormal patterns in the data. The AGRU model is trained on labeled data, distinguishing between normal and anomalous instances.
- The trained AGRU model is evaluated using appropriate performance metrics such as accuracy, precision, recall, and F1-score. The model's performance is assessed on both training and testing datasets to ensure its generalization ability. If necessary, the

model can be refined via hyper parameter tuning or employing ensemble techniques to further improve its anomaly detection capabilities.

Section 2 represents the associated work, including the table that explains the benefits and drawbacks of the current methodologies. Section 3 shows the proposed methodology and discusses the proposed model with equations, figures, etc. Section 4 shows the findings from a comparison of the proposed model to existing models using graphs, tables, etc. The conclusion, which includes future scope and references, is found in Section 5.

2. Related Works

The existing methods used to enhance the agricultural domain are discussed below.

IoT-driven data mining for smart crop prediction in the peasant agricultural sector was presented by Mendoza et al. [21]. The approach makes use of Internet of Things (IoT) sensors to record agricultural characteristics and collect information from peasants via mobile applications about the volume of crop production. The solution incorporates data storage services, IoT message hubs, ML models, and IoT data analytics along with data mining concepts. The method's advantage is that it makes use of less expensive IoT sensors, data analysis services, and well-known data storage devices. The method's drawback is that the system requires routine maintenance. An ML-based strategy for precise agriculture in 5G-based IoT was presented by Murugamani et al. [22]. The proposed approach involves assessing soil quality and looking for diseases in cotton leaves. Using regression-based techniques, cotton leaf disease can be identified and classified. The farmer received information about the crop's infection via a mobile application. Raspberry Pi served as an interface for four different kinds of sensors. The process was inexpensive. Due to the presence of big particles in sandy soil, the method's accuracy was limited. The idea of the detection and classification of intrusions into IoT networks used in agriculture was put forth by Raghuvanshi et al. [23]. By using principal component analysis, features from the pre-processed NSL-KDD dataset were extracted. The pre-processed data were then classified using ML methods such as linear regression, support vector machine (SVM), and random forest. The method's benefit is that it makes sure that the IoT network is secure. The method's low categorization accuracy is a drawback. Kethineni et al. [24] developed a powerful intrusion detection system for smart agriculture DDoS attack detection. Data normalization and label encoding were used to pre-process the gathered data. The CNN algorithm was merged with the Bi-GRU model, which can both detect and categorize intrusions. The attention mechanism in the BiGRU model looks for the most important characteristics that can be used to recognize the DDoS attack. The wild horse optimization algorithm was also used to improve the model's classification accuracy. Nevertheless, only some IoT attacks were detected using this method. Ferrag et al. [25] introduced CNNs, deep neural networks, and recurrent neural networks. The effectiveness of deep learning and machine learning methods for cyber-security in agriculture 4.0 was specifically evaluated. Using two brand-new real traffic datasets, the CICDDoS2019 dataset and the TON IoT dataset, the efficiency of each model was examined for both binary and multi-class classifications. For multiple-dimensional time series data produced using the smart agricultural Internet of Things, Cheng et al. [26] suggested a GAN-based anomaly detection algorithm. For the purpose of identifying anomalies, the model employed reconstruction techniques after learning the distribution patterns of typical data using the GAN structure. An upgraded improved LSTM network was taken into consideration to serve as the foundation for the GAN because time series data have temporal dependence and there may be a potential correlation between multiple variables. The encoder–decoder was chosen as the GAN generator to address the generator inversion problem. By doing this, the requirement for real-time anomaly detection was effectively addressed while the computation time was also decreased. However, the time window size option was the most important factor to be taken into account; even if the time window was extended, the model's performance decreased. A serial–parallel convolutional bidirectional gated recurrent network model incorporating ensemble classifiers was a flexible and systematic hybrid model proposed

by Zhang et al. [27]. In terms of identifying smart contract vulnerabilities, the model demonstrated excellent performance gains. A serial–parallel convolution (SPCNN) was also suggested as a viable option for a hybrid model. By preserving the temporal structure and geographic location data, it extracts characteristics from the input sequence for multivariate combinations. The model's resilience increased throughout the classification stage by using the ensemble classifier. However, the technique struggles to detect integer overflow vulnerabilities. A more scalable method of identifying smart contract vulnerabilities was presented by Zhang et al. [28] using a multiple-objective detection neural network (MODNN). Without specialized or predetermined information, MODNN can validate 12 types of vulnerabilities, including 10 known risks, and find additional unknown types using implicit characteristics and multi-objective detection (MOD) techniques. Because of its great scalability and support for simultaneous vulnerability detection, it does not require separate models to be trained for different types of vulnerabilities, saving both a lot of time and money. The absence of smart contract vulnerability datasets was also addressed in this work by the creation of a smart contract-crawler (SCC). An experimental review of neural-based approaches for network intrusion management was presented by Di Mauro et al. [29]. A critical comparison of approaches shares a common paradigm including intrusion detection, such as weightless neural networks. Then, the CIC-IDS-2017/2018 dataset was utilized, wherein single-class cases (benign vs. malign) and multi-class cases (n benign vs. malign1... vs. malignk) were deemed to be important. The network abnormal traffic detection model based on semi-supervised deep reinforcement learning was presented by Dong, S et al. [30]. A semi-supervised double deep Q-network (SSDDQN)-based optimization model for network anomalous traffic identification was dependent on the double deep Q-network. In SSDDQN, the current network adopts the auto encoder to rebuild the traffic features. A deep neural network was applied as a classifier. The NSL-KDD and AWID datasets were used for training with testing and made for a comprehensive comparison. Deep learning for the categorization of Sentinel-2 image time series was presented by Pelletier, C. et al. [31]. The two leading deep learning approaches for handling temporal data, recurrent neural networks and temporal convolutional neural networks, were used Rangwani, D et al. [32], Vangala, A et al. [33], Vidyashree et al. [34]. For precise and current land cover mapping across huge areas, satellite image time series have shown to be indispensable Chen, M et al. [35], Gutpa, A et al. [36], Bakthavatchalam et al. [37]. The majority of SITS-related writings have emphasized the use of conventional classification techniques such as random forests Colombo-Mendoza, L.O et al. [38], Murugamani, C et al. [39], Raghuvanshi, A et al. [40].

However, with the increasing adoption of these technologies, there are several security issues that need to be considered to prevent privacy leaks and other potential consequences. A major security issue in smart agriculture is the risk of data breaches. Farmers and agricultural companies collect and store sensitive data on crop yields, weather patterns, and soil conditions, which can be valuable to cyber-criminals. If these data fall into the wrong hands, they can be used to manipulate markets, cause crop failures, and other harmful activities. Another issue is the vulnerability of the devices and networks used in smart agriculture. These devices, such as sensors and drones, are often connected to the Internet and can be hacked, leaving the entire system open to attacks. Malware and other malicious software can also be introduced into the system through these devices, compromising sensitive data. The consequences of privacy leakage in smart agriculture can be significant. For instance, if the data collected from smart agriculture systems were to be leaked, it could lead to decreased confidence in the industry and ultimately a decline in the adoption of these technologies. Additionally, farmers could face serious financial losses if their sensitive data, such as crop yields or pricing data, were to fall into the hands of competitors or malicious actors. Overall, it is crucial to ensure that smart agriculture systems are designed with security and privacy.

A comparison with the state-of the art techniques is shown in Table 1.

Table 1. Comparison with state-of the art techniques.

Author and Year	Aim	Merits	Demerits
Mendoza et al., 2022 [21]	IoT-driven data mining for smart prediction of crops in the domain of peasant farming	It uses cheaper IoT sensors, data analytical services, and popular data storage devices	The system needs periodic service
Murugamani et al., 2022 [22]	ML-based approach for precise agriculture in 5G-based IoT	Affordable	In sandy soil, the accuracy is lower due to the presence of large particles
Raghuvanshi et al., 2022 [23]	Detection and classification of intrusions into IoT networks utilized in agriculture	It ensures security in IoT network	Classification accuracy is low
Kethineni et al., 2023 [24]	Powerful intrusion detection system for smart agriculture DDoS attack detection	Accurate detection	Only some IoT attacks are detected
Ferrag et al., 2021 [25]	For cyber-security in agriculture 4.0	Examined for both binary and multi-class classifications	Complexity while using
Cheng et al., 2022 [26]	Multiple-dimensional time series data produced using smart agricultural Internet of Things	GAN because time series data have a temporal dependence and there may be potential correlation between multiple variables	Even if the time window is extended, the model's performance is going to decrease
Zhang et al., 2022 [27]	A flexible and systematic hybrid model	Identifying smart contract vulnerabilities, the model demonstrated excellent performance gains	Technique struggles to detect integer overflow vulnerabilities
Zhang et al., 2022 [28]	Identifying smart contract vulnerabilities	Its great scalability and support for simultaneous vulnerability detection; it does not require separate models to be trained for different types of vulnerabilities	The system may be complex in nature
Di Mauro et al., 2020 [29]	Experimental review of neural-based approaches for network intrusion management	Depends on time complexity and the feed-forward incremental learning activated in WiSARD	Neural networks require large amounts of labeled data to be trained effectively
Dong, S et al., 2021 [30]	Network abnormal traffic detection model depends on semi-supervised deep reinforcement learning	Improved efficiency	Difficulty with adversarial attacks and limited interpretability
Pelletier, C. et al., 2019 [31]	Categorization of Sentinel-2 image time series	High accuracy in image classification and improved generalization	Requires significant computational resources to be trained and operated, which can be expensive and time-consuming

3. Proposed Methodology

Because of the development and widespread usage of Internet of Things (IoT) devices, precision agriculture is revolutionizing traditional farming methods. The Internet is an open channel that the agricultural and IoT industries utilize to help farmers to collect, process, monitor, and make informed decisions about their farms. However, using the Internet exposes users to a variety of risks, including security risks (such as carrying out cyber-attacks), data privacy risks (such as inference attacks and data poisoning), etc. The use of traditional centralized security methods has drawbacks in terms of scalability, verifiability,

traceability, and single point of failure. We proposed a privacy-encoding-based enhanced deep learning framework that uses an intrusion detection model, transformation technique, and perturbation-based encoding. The secrecy is considered to be the unprocessed data produced via IoT-based smart agriculture in this study. The four steps of privacy were used here. Feature selection, feature mapping, feature normalization, and feature encoding were used for a normal strategy, along with intrusion detection. The feature selection method uses processing time and offers more dependable and superior privacy. Data are changed into a new encoded format using the sparse capsule-auto encoder approach to make them more resistant to attacks. In order to detect intrusions, attention-based gated recurrent unit neural networks are used with encoded data. To demonstrate the effectiveness of the suggested framework for recognizing normal and attack patterns, the ToN-IoT dataset and IoT Botnet were used to test and train the proposed method. The whole framework is depicted in Figure 1.

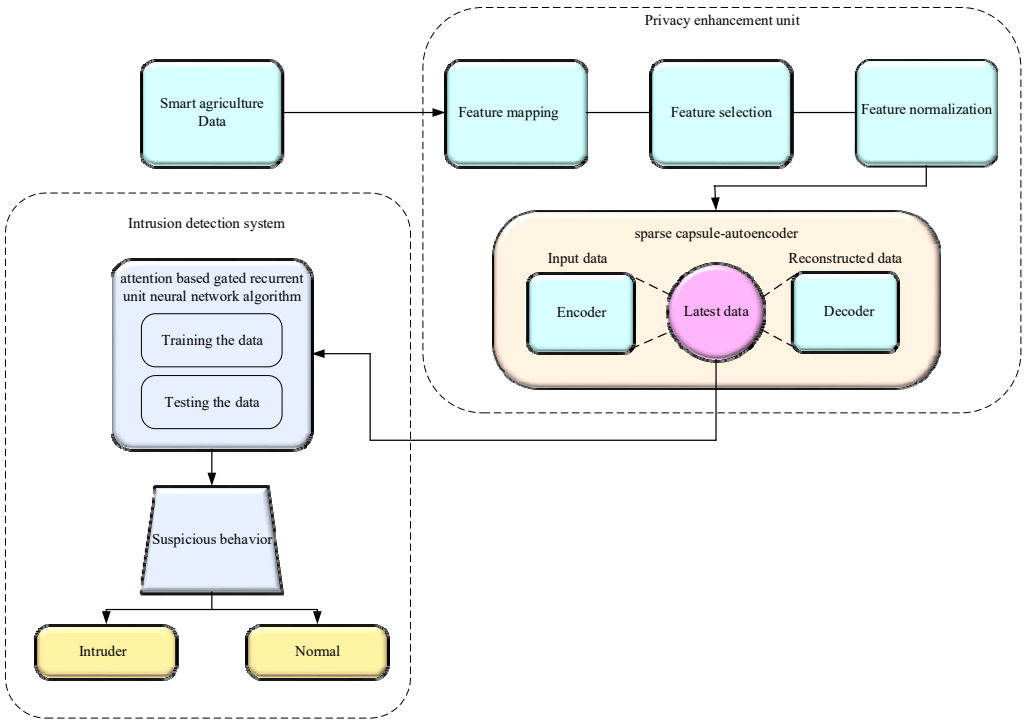

Figure 1. Framework of proposed model.

3.1. Strategies for Detecting Intrusions

A piece of equipment or program known as the intrusion detection system (IDS) monitors online activity and guards authorized network users against illegal actions that risk availability, integrity, and security. The intrusion detection system can be generically categorized according to two factors: (I) installation and (II) detection method. Host-based intrusion detection systems (HIDSs) and network-based intrusion detection systems (NIDSs) are the first two categories into which IDS-basis deployment is split. The host-based intrusion detection scheme was created in the host computer (device/client). Internal invasion is less likely with HIDS, while NIDS records and examines the movement of detected packets. Second, there are two categories of IDS-based detection techniques: (I) knowledge- or signature-based (SIDS) and (II) anomaly-based (AIDS). To identify an active attack, SIDS uses an authentication system. Although SIDS has a high probability of recognition, it cannot detect newly occurring disruptive events in the network. In contrast, in AIDS, typical computer system behavior is represented using statistical, machine learning, or knowledge-based techniques. When AIDS observes a discrepancy between the proposed model behavior and the actual behavior, the discrepancy is interpreted as an intrusion.

3.2. Feature Mapping

IoT feature values can be found in both categorical and numerical forms in network traffic. Categorical values are transformed into numerical values using the mapping approach. For instance, the ToN-IoT dataset's protocol feature value is transformed into ordered numbers such as 1, 2, etc. This is due to the privacy-preservation mechanism's ability to handle numerical attributes, as seen below.

3.3. Feature Selection

To find the most diverse characteristics for the available dataset, the feature selection procedure was used. Better characteristics increase the likelihood of accurate classification. Reducing duplicate and repeated components from IoT network traffic accelerates processing while improving detection techniques and privacy. The proposed model chooses its features based on a reciprocal information-based process. The quantity of information gained from the feature f_1 through the feature f_2 is approximately calculated between the two random characteristics f_1 and f_2. Using the below equation, the mutual information (M_I) function was computed (1).

$$M_I(f_1; f_2) = \int_{f_1}^{\int \frac{P(f_1,f_2)}{P(f_1)P(f_2)}_{12}} \int_{f_2}^{\int} P(f_1, f_2) log \tag{1}$$

where the general probability function is given by $P(f_1, f_2)$ of function f_1 and f_2.$P(f_1)$ and $P(f_1)$ are the marginal density functions. The mutual information between the target variable f_2 and the subset of the selected features Z will be optimized when selecting a feature.

$$\overline{M} = \underset{M}{argmax} U(Z; f_2), s.t. |M| = X \tag{2}$$

X is the total number of characteristics that were chosen and employed in the model's development. To choose the most advantageous number of features, each of the aforementioned processes were carried out with respect to the training dataset.

3.4. Feature Normalization

This technique transforms feature data to a common scale between 0 and 1. The maximum value of a feature is changed to 1, while the smallest value is transformed to 0. Deep learning methods' training times are sped up in this way. The min–max normalization method is employed in privacy-encoding-based enhanced deep learning architecture.

$$N_f = \frac{f - f_{min}}{fmin_{max}} \tag{3}$$

f denotes the feature that needs to be shrunk and N_f denotes the new feature; f_{max} represents the highest value and f_{min} defines the lowest value for each given feature.

3.5. Feature Encoding

The auto encoder is used to encrypt the feature normalized data. The sparse capsule-auto encoder (SCAE) is the suggested auto encoder which is discussed in the following section.

3.5.1. Auto Encoder (AE)

An auto encoder is a three-layer symmetric neural network that extracts the relevant features from unlabeled high-dimensional complex input utilizing system design that has been enhanced via unsupervised layer-by-layer greedy learning. A typical AE's construction is depicted in Figure 2. Encoding and decoding refer to the operations that are performed from the hidden layer to the output and from the input to the hidden layer, respectively. The decoding reconstructs the input data under the extracted features after the encoding has extracted features from the dataset.

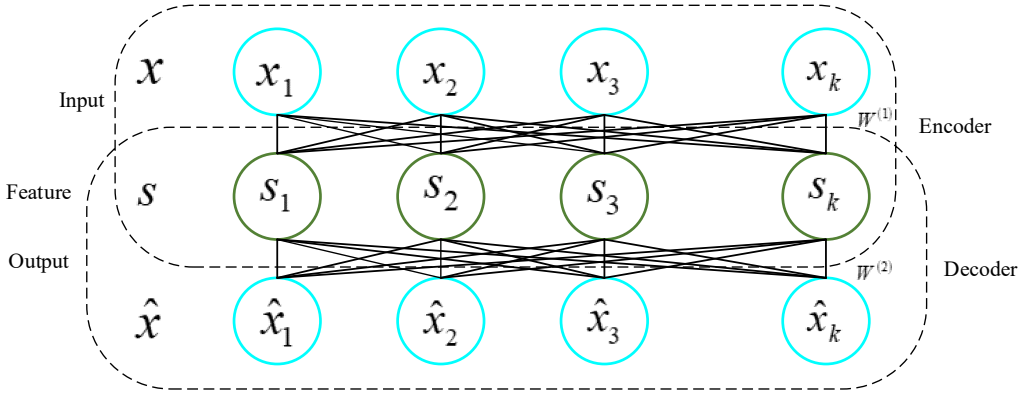

Figure 2. The structure of an auto encoder.

In order to create l significant features $x = [x_1, x_2, x_3, \ldots x_k]^C \in R^{k\times 1}$ from an unlabeled training sample $s = [s_1, s_2, s_3, \ldots s_l]^C \in R^{l\times 1}$ using the whole connection layer, the input is translated by the encoder to the hidden state. Then, using the same technique, the decoder creates the output $\hat{x} = [\hat{x}_1, \hat{x}_2, \hat{x}_3, \ldots \hat{x}_k]^C \in R^{k\times 1}$ from the feature s. Equations (4) and (5) are descriptions of the encoding and decoding processes, respectively:

$$s = f(x) = \Phi_{af}\left(W^{(1)}x + b_a\right) \tag{4}$$

$$\hat{x} = g(s) = \Phi_{af}\left(W^{(2)}s + b_b\right) \tag{5}$$

where $W^{(1)}$ and $W^{(2)}$ are the entire connection layer weight matrices, b_a and b_b are the full connection layer bias matrices, and Φ_{af} activates a nonlinear activation process made up of a number of different types, such as the ReLU, TanH, and sigmoid.

The intrinsic characteristics S of the input are acquired as the reconstruction loss in the parameter set, which is constantly minimized $\left\{W^{(1)}, b_a, W^{(2)}, b_b\right\}$. The mean square error (MSE) loss function and cross-entropy (CE) loss function are frequently used to calculate the reconstruction error among the input vector x and reconstruction vector $\hat{x}$. For a single training dataset $x = [x_1, x_2, x_3, \ldots x_k]^C \in R^{k\times 1}$, the CE loss function is exhibited in (6), and the MSE loss function is exhibited in (7):

$$D_{ce} = -\sum_{i=1}^{k}[x_i log(\hat{x}_i) + (1 - x_i) log(1 - \hat{x}_i)] \tag{6}$$

$$D_{mse} = \frac{1}{2k}\sum_{i=1}^{k}\|\hat{x}_i - \hat{x}_i\|^2 \tag{7}$$

3.5.2. Capsule Auto Encoder

The capsule auto encoder is used in addition to the sparse auto encoder to improve efficiency and strengthen attack prevention because the sparse auto encoder cannot defend the smart agriculture database by itself. The conventional capsule auto encoder has two main modules which are the object capsule auto encoder (OCAE) and part capsule auto encoder (PCAE). To begin, the PCAE employs a CNN to extract the pose, presence, and features of each component of the objects in the input image. After encoding the scattered components produced by the PCAE into bigger objects using a set transformer, the OCAE outputs the presence probabilities of all of the potential objects. The classifier then outputs a classification outcome based on the output of the OCAE. Figure 3 illustrates the CAE architecture, which is comparable to that of the normal AE but uses a different calculation method.

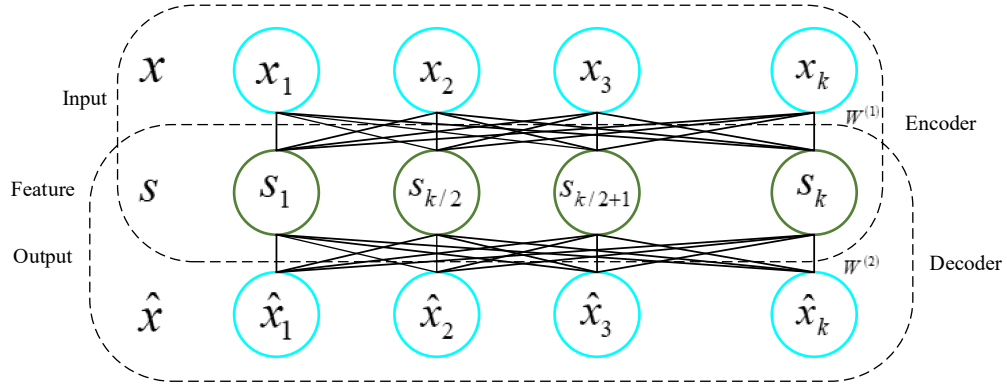

Figure 3. Structure of CAE.

The encoder translates the input into a feature layer to produce the feature set $s = \left[s_1^C, s_2^C\right]^C = [[s_1, \ldots s_{k/2}], [s_{k/2+1}, \ldots, s_k]]^C \in R^{k\times 1}$ for an unlabeled training sample $x = [x_1, x_2, x_3, \ldots x_k]^C \in R^{k\times 1}$, and the decoder subsequently produces the output $\hat{x} = [\hat{x}_1, \hat{x}_2, \hat{x}_3, \ldots \hat{x}_k]^C \in R^{k\times 1}$. Equations (8) and (9) can be used to explain the encoding and decoding methods:

$$s = \left[s_1^C, s_2^C\right]^C = f(x) = \Phi_{af}\left(W^{(1)}x + b_a\right) = \Phi_{af}\left(\left[W_1^{C(1)}, W_2^{C(1)}\right]^C + \left[b_{W_1^{(1)}}, b_{W_2^{(1)}}\right]^C\right) \quad (8)$$

$$\hat{x} = g(s) = \Phi_{af}\left(W^{(2)}s + b_b\right) = \Phi_{af}\left(W^{(2)}\left[s_1^C, s_2^C\right]^C + b_{W^{(2)}}\right) \quad (9)$$

where $W^{(1)} \in R^{k\times k}$ and $W^{(2)} \in R^{k\times k}$ are their weight matrices and $b_{W^{(1)}} \in R^{k\times 1}$ and $b_{W^{(2)}} \in R^{k\times 1}$ are the entire connection layers' bias vectors. The bias vectors and weight matrices for the vector feature s_1 are $W^{(1)} \in R^{(k/2)\times k}$ and $b_{W_1^{(1)}} \in R^{(k/2)\times 1}$, respectively. The weight matrices along the bias vectors of vector feature s_2 are $W^{(1)} \in R^{(k/2)\times k}$ and $b_{W_2^{(1)}} \in R^{(k/2)\times 1}$, respectively.

Both CAE and conventional AE go through the same training process. As a result, by minimizing the loss function, the ideal parameter set $\left\{W^{(1)}, b_{W^{(1)}}, W^{(2)}, b_{W^{(2)}}\right\}$ and also the intrinsic vector features are achieved. As demonstrated in (10), the MSE loss function is employed in this study.

$$D_{mse} = \frac{1}{2k}\sum_{i=1}^{k}\|\hat{x}_i - \hat{x}_i\|^2 = \frac{1}{2k}\sum_{i=1}^{k}\|\hat{x}_i - \Phi_{af}\left(W^{(2)}\Phi_{af}\left(W^{(1)}x + b_{W^{(1)}}\right) + b_{W^{(2)}}\right)\|^2 \quad (10)$$

CAE can extract two vector features $\{s_1, s_2\}$ during training, whereas normal AE can only extract l scalar features $\{s_1, s_2, s_3 \ldots s_l\}$. Vector features have been shown to exceed scalar features in classification situations.

3.5.3. Sparse Capsule-Auto Encoder (SCAE)

After choosing the features using the MI-based feature selection procedure, the reduced features are extracted from the provided raw data utilizing the SCAE technique. In SCAE, the original data are transmitted over a weight link to the hidden layer. To recreate the results, the hidden layer's activation value is mapped to an output layer. By carefully changing the weight that is meant to provide proper data representations, the reconstruction error is reduced. The total number of concealed nodes is widely distributed to prevent excessive node activation. Additionally, the sparse penalty restriction is included to restrict "active" neurons, and if a neuron's output equals 1, the auto encoder (AE) transforms into a SCAE during application; otherwise, the neuron is referred to as "inactive". Consider the dataset $DS_i = [ds_{ij}, ds_{22}, \ldots ds_{mn}]$, where $i = 1, \ldots m$ indicates a number of samples

and $j = 1, \ldots n$ indicates the dimension of samples. Using Equation (11), it is possible to determine the activation h_l of the hidden layer node.

$$h_l = af_l\left(b^{(1)} + W^{(1)}dt\right) \tag{11}$$

where $W^{(1)}$ stands for the weight connecting the input and hidden layers and $b^{(1)}$ represents the bias. The "elu" function, which represents the activation function, is employed. To decode the hidden layer and obtain original data, Equation (12) determines the weight relationship between both the output and hidden layer.

$$\hat{r} = af_l\left(b^{(2)} + W^{(2)}h_l\right) \tag{12}$$

$\hat{r}$ stands for reconstructed data. $W^{(2)}$ stands for the weight among the output and hidden layer, while $b^{(2)}$ stands for the bias.

In AE, minimizing the reconstruction error R_l, which may be calculated as variance between the input data d_t and reconstructing data $\hat{r}$, the parameter set is indicated through $\varepsilon = \left\{W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}\right\}$. Using Equation (13), the, d_i,$i = 1, \ldots m$ cost functioning of the auto encoder is calculated for the dataset.

$$H_E(\varepsilon) = \sum_{i=1}^{m} R_l\left(d_{t(i)}, \hat{r}_i\right) \tag{13}$$

The hidden unit's mean activation is $H = 1 \ldots j_l$, which, in the context of SCAE, can be calculated using Equation (13), and it is pre-summated to be near to '0', indicating that the bulk of neurons in the hidden layer are "deactivated". To penalize ϑ_k, the sparse penalty is presented if the established sparse parameter ϑ significantly deviates from it. When determining the difference between the derived sparse parameter ϑ_k and ϑ using Equation (14), Kullback–Leibler (KL) divergence is applied.

$$\vartheta_k = \frac{1}{m}\sum_{i=1}^{m}(af_l(b_k + W_{ik}d_{ti})) \tag{14}$$

$$\sum_{k=1}^{j_l} PQ(\vartheta \| \vartheta_k) = \sum_{k=1}^{j_l} \vartheta log\frac{\vartheta}{\vartheta_k} + (1-\vartheta)log\frac{(1-\vartheta)}{(1-\vartheta_k)} \tag{15}$$

Nevertheless, Equation (16) can be used to define the total cost function and sparse constraint for SCAE.

$$H_{SE}(\varepsilon) = \sum_{i=1}^{m} R_l\left(d_{t(i)}, \hat{r}_i\right) + \sum_{k=1}^{j_l} PQ(\vartheta \| \vartheta_k) \tag{16}$$

where $H_E(\varepsilon)$ and the ε parameter are tuned for the sparse penalty. The back propagation algorithm's original weight and bias are updated to resolve the partial derivative of H_{SE} to ε and lower the cost function. A trained SCAE network is acquired when the parameter has been upgraded. The product of the existence probability of part capsules and the value of the learned alpha channel for each determine the mixing probabilities of various components. The pre-training procedure in the suggested method aids SCAE in learning many nonlinear transformations, capturing the major changes in frequency spectra. After nonlinear transformation, the feature vector is directly truncated to ensure that the two features are distinct from one another and that the following features are diverse. The SCAE is then assisted in finding the substantially discriminative data from various stages throughout the fine-tuning phase, thus enhancing the independence of the generated vector features. The hidden layer's activation value is seen as an extracted feature.

3.6. Intrusion Detection Using AGRU

In the proposed approach, the intrusion detection stage is crucial. An attention-based AGRU (Algorithm 1) was employed for detection. The following section gives a more detailed examination.

Attention-Based Gated Recurrent Unit Neural Network Algorithm (AGRU)

A traditional feed-forward neural network has evolved into a recurrent neural network (RNN). The data flow in the conventional neural network paradigm is unidirectional, that is, starting with the input layer and then moving via the hidden layer to the output layer. Figure 4 depicts the reset gate R_i and update gate U_i, the two gate structure components that make up the GRU. The current moment set of the hidden states is where the flow of information about the hidden states from the previous moment in the current set is controlled by the R_i gate. The U_i gate is used to govern how much irrelevant content from the previous state has to be forgotten and to decide the amount of the hidden state that should be kept.

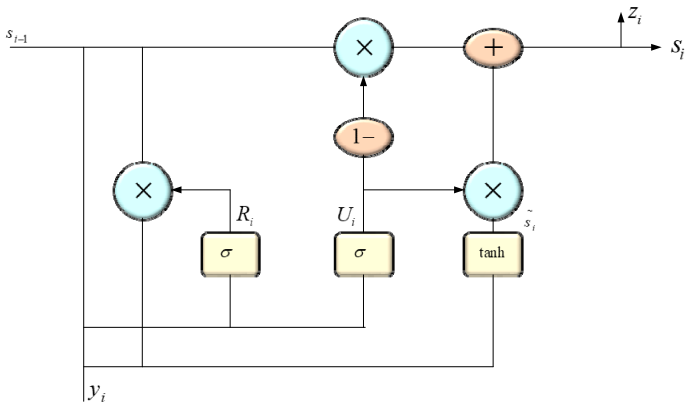

Figure 4. Gated recurrent unit neural network.

In this study, we utilized R_i to signify the reset gate and U_i to denote the update gate, as illustrated in Figure 4. After that, the GRU learning model can be explained as follows.

First, in a GRU, past information (s_i) and present information (y_i) define the reset gate and update gate.

$$R_i = \sigma(W_R \bullet [s_{i-1}, y_i]) \tag{17}$$

$$U_i = \sigma(W_U \bullet [s_{i-1}, y_i]) \tag{18}$$

Algorithm 1: AGRU

Reset gate and update gate are given as follows:
$R_i = \sigma(W_R \bullet [s_{i-1}, y_i])$ (17)
$U_i = \sigma(W_U \bullet [s_{i-1}, y_i])$ (18)
The reset gate, which controls a GRU's candidate set, operates according to the following formula:
$\widetilde{s_i} = tanh(W_s \bullet [R_i \times s_{i-1}, y_i])$ (19)
Third, the GRU updates s_i using the following formula during the update memory phase:
$s_i = (1 - U_i) \times s_{i-1} + U_i \times \widetilde{s_i}$ (20)
The forward propagation's final result is z_i which can be calculated using the following formula:
$$z_i = softmax(W_0 * s_i)$$

The reset gate, which controls a GRU's candidate set, operates according to the following formula:

$$\widetilde{s_i} = tanh(W_s \bullet [R_i \times s_{i-1}, y_i]) \tag{19}$$

Third, the GRU updates s_i using the following formula during the update memory phase:

$$s_i = (1 - U_i) \times s_{i-1} + U_i \times \widetilde{s_i} \tag{20}$$

The forward propagation's final result is z_i, which can be calculated using the following formula:

$$z_i = softmax(W_0 * s_i) \tag{21}$$

The ability to concentrate on certain items due to attention in the human visual system can significantly increase how successfully people process visual information. Attention is a method that simulates cognitive attention in neural networks. It is a well-liked approach in the disciplines of machine translation and recommendation systems, among others. Deep neural networks may more quickly latch onto key areas of the objective from an amount of data, eliminate irrelevant data, and work more efficiently by incorporating attention processes. The attention mechanism and deep learning model can be compared to a weight matrix. To determine the proper attention level, the associated weight value for each input value is multiplied by the input value. The weight factor can be increased by maximum attention.

In the multi-channel time series prediction task, the ultimate objective is to predict r_i at the subsequent time based on the physical values from g_1 to g_i, previously observed in the time window j. Nevertheless, more complex temporal properties are seen in time series data sensor information. Different time steps have varying degrees of success in foretelling future data. In order to acquire pertinent information to aid in the prediction of the future physical value $\hat{r}_i$, we applied soft attention to derive a context vector c_i. The following formula determines the attention mechanism:

$$\beta_t = \frac{exp(S_t)}{\sum_{t=1}^{i=1} exp(S_t)} \tag{22}$$

At each time step, we calculated the associated weight score, and then applied the softmax function to normalize the score to generate the distribution of conditional probabilities β_t, which indicates the relevance of the t^{th} time window for prediction. The function calculates the score S_t.

$$S_t = tanh\Big(a^l hs_t + b\Big) \tag{23}$$

where a and b are learning parameters that may be trained along with other model parameters. The context vector v_i is calculated by the attention mechanism as the weighted average of all of the concealed states of hs_1 to hs_{i-1}.

$$v_i = \sum_{t=1}^{i=1} \beta_t hs_t \tag{24}$$

We integrated the data from the two vectors, the context vector v_i, and the current hidden state hs_i to produce an attentional hidden state, as shown below:

$$k* = tanh(w_k[v_i; hs_i]) \tag{25}$$

To avert overfitting, the attentional vector $k*$ is supplied into the dropout layer. The dense layer predicts the following data.

Time complexity for attention-based gated recurrent unit neural network.

This depends upon the number of input features, the number of time steps, the size of the hidden layer, and the number of attention heads.

Assuming that the input has N features, the hidden layer size is H, the number of time steps is T, and the number of attention heads is A, the time complexity of an attention-based GRU can be expressed as $O(TAH\hat{}2 + TH\hat{}2)$.

The first term in the equation, $TAH\hat{}2$, corresponds to the computational cost of computing the attention weights. The attention mechanism involves computing a weighted sum of the hidden states from each time step, and this operation needs to be performed for each attention head. The complexity of computing the attention weights for one head is $O(TH\hat{}2)$, and since there are A attention heads, the total complexity for computing the attention weights is $TAH\hat{}2$ The second term in the equation, $TH\hat{}2$, corresponds to the computational cost of updating the hidden state using the GRU cell. This operation involves matrix multiplications and element-wise operations, and it needs to be performed for each time step. The complexity of updating the hidden state for one time step is $O(H\hat{}2)$, and since there are T time steps, the total complexity for updating the hidden state is $TH\hat{}2$. Overall, the time complexity of an attention-based GRU neural network is $O(TAH\hat{}2 + TH\hat{}2)$. This complexity can be reduced by using techniques such as pruning, quantization, or parallelization, which can help to speed up the computations and reduce the memory requirements of the network.

Using an attention mechanism in combination with the recurring neural network for the identification of an intrusion detection task further complicates and improves the model.

4. Results and Discussion

To validate the efficacy of the proposed technique, the performance indicators of accuracy, detection rate, precision, f-score, testing and training, validation accuracy and validation loss, confusion matrix analysis, and RoC curve were examined and compared with the existing techniques. The implementation took place using Python and the results were tested and plotted.

Python programming was employed in the creation of the PEDL framework. The Intel(R) Core(TM) i5-3570 CPU clocked at 3.40 GHz 3.40 GHz, together with 8 GB of installed RAM, powers the SST003.seahost.local computer. Two datasets based on IoT, which are IoT-Botnet and ToN-IoT, were utilized to evaluate the effectiveness of the intrusion detection systems (IDs). These were chosen because they offer a variety of security procedures and reliable IoT network results. To train and assess how well the PEDL technique distinguishes between attack and typical observations, these datasets were divided into testing and training sets. The results were compared using the deep neural network, convolutional neural network, recurrent neural network, and long short-term memory. The hyper parameters are given in Tables 2 and 3, representing the system configuration.

Table 2. Hyper parameters.

Hyper Parameters	
No. of hidden layers	10
No. of hidden nodes	60
Dropout rate	0.2
Optimizer	adam
Activation	relu
No. of epoch	20
Batch size	121
Loss	Mean square error

Table 3. System configuration.

System Configuration	
Device name	SST003
Processor	Intel(R) Core(TM) i5-3570 CPU @ 3.40 GHz 3.40 GHz
Installed RAM	8.00 GB (7.89 GB usable)
Device ID	330431F3-8552-4664-BCAD-E0108D59137B
Product ID	00330-80000-00000-AA440
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display
Edition	Windows 10 Pro
Version	21H2
OS build	19044.2728

4.1. Datasets Used to Assess the Efficiency of the PEDL Structure

Two of the latest IoT-based publicly accessible datasets were utilized to confirm the efficacy of the suggested PEDL approach. The training and test sets were divided into 70% and 30%, respectively. Below is a description of each dataset's specifics:

ToN-IoT: The ToN-IoT dataset holds a variety of information gathered from IoT and IoT sensor telemetry records. This dataset includes a variety of current attack instances that have been discovered in IoT contexts, viz, Backdoor, MITM, DDoS, DoS, Injection, Password, Scanning, XSS, and Ransomware. A normal vector, nine different sorts of attacks, and 43 labeled features make up this dataset. These features are used to detect the external attack for the smart agriculture. The model utilizes 378,782 data points from this dataset, which are later separated into 303,025 and 75,757 for training and testing purposes. Table 4 tabulates common and unique attack event statistics.

Table 4. The training and testing details of the dataset.

Dataset	Partition of Class	Entire Occurrences	Rate of Class Frequency (%)
ToN-IoT training set	Backdoor (0)	14,135	4.35
	DDoS (1)	13,971	4.32
	DoS (2)	13,913	4.33
	Injection (3)	14,071	4.34
	MITM (4)	727	0.21
	Normal (5)	209,792	65.06
	Password (6)	14,017	4.37
	Ransomware (7)	13,992	4.32
	Scanning (8)	14,100	4.29
	XSS (9)	14,012	4.36
ToN-IoT testing set	Backdoor (0)	5865	4.29
	DDoS (1)	6029	4.36
	DoS (2)	6087	4.33
	Injection (3)	5929	4.31
	MITM (4)	316	0.25
	Normal (5)	90,208	65.07
	Password (6)	5983	4.26
	Ransomware (7)	6008	4.36
	Scanning (8)	5900	4.44
	XSS (9)	5988	4.28

Table 4. *Cont.*

Dataset	Partition of Class	Entire Occurrences	Rate of Class Frequency (%)
IoT Botnet training set	DDoS (0)	455,786	33.56
	DoS (1)	445,578	32.80
	Reconnaissance (2)	388,508	28.60%
	Normal (3)	68,038	5.01%
	Theft (4)	364	0.03%
IoT Botnet testing set	DDoS (0)	73,602	33.59%
	DoS (1)	71,971	32.85%
	Reconnaissance (2)	62,477	28.51%
	Normal (3)	11,015	5.03%
	Theft (4)	48	0.02%

IoT Botnet: The above was modified to create the IoT Botnet dataset. Only a few flow features, out of 46 network features, are present in BoT-IoT. The BoT-IoT flow characteristics and network features are multiplied and optimized using a network traffic flow detector. This dataset contains a standard vector, four alternative attack types, and 88 tagged features. This dataset contains 50,000 data points that are separated into 40,000 and 10,000 for training with testing. These properties are used to train the model to detect an attack and also enable effective communication for smart agriculture. The statistics of common and unique assault occurrences in the dataset are depicted in Table 4.

4.2. Description of Evaluation Metrics

To determine the effectiveness of the IDS, multiple performance evaluation matrices were employed. Recall, accuracy, F1-score, and precision were some of these measurements. Nevertheless, many parameters that were utilized to calculate these ratings are explained below.

The efficacy of the IDS in distinguishing between invasive and nonintrusive activities is described by the metrics in this area. An IDS, which is a binary classifier, may provide one of the following outcomes: true positive (p_t) refers to the accurate classification of an intrusion; true negative (n_t) refers to the accurate classification of a reasonable action; false positive (p_f) refers to the incorrect classification of a reasonable action as an intrusion; and false negative (n_f) refers to the incorrect classification of an intrusion. The well-known measures are given below.

Accuracy: This measurement is simply the percentage of validation sets or test sets that an IDS correctly classifies. Accuracy is achieved using the following:

$$Accuracy = \frac{p_t + n_t}{p_t + n_t + p_f + n_f} \tag{26}$$

Precision: This indicator shows how often the IDS's classified activities are intrusive. Precision is achieved using the following:

$$precision = \frac{p_t}{p_t + p_f} \tag{27}$$

Recall: This statistic measures the proportion of invasive behaviors that the IDS deems to be intrusive. The recall is also known as the detection rate (DR). Detection rate data are acquired via the following:

$$Recall = \frac{p_t}{p_t + n_f} \tag{28}$$

F1-score: This measure is a weighted harmonic mean of memory and precision, which represents the importance of recall with respect to accuracy. When assessing a multi-class classification, the F-score is also used. Equation (29) provides the F1-score. By micro-averaging using the class frequency (mico-F1) or macro-averaging based on the identical relevance of all classes, the final F1-score is achieved (macro-F1). When there is an unbalance in the classes, binary and multi-class classifiers are evaluated using the F1-score in comparison to the G-measure, which is the geometric mean of the detection rate and accuracy.

$$F1 - score = 2 * \frac{DR \times precision}{DR + precision} \tag{29}$$

4.3. Evaluation of Intrusion Detection System

Each dataset's twenty characteristics are subjected to the attention-based GRU approach for confidentiality. This technique shields data against inference attacks that may gather private model information. Including both datasets, the attention-based GRU was built up using the 20 characteristics listed in Table 5 as the input layer. One output node, five hidden nodes, and an Elu activation function make up the encoder. The final model was set up to predict attacks and normal classes using the following parameters: optimizer = Adam, epochs = 10, batch size = 100, and dropout = 0.1.

Table 5. Statistics of the features chosen using the two datasets.

Dataset	Selected Features	Overall Features Selected
ToN IoT	20	F1, F2, F5, F8, F10, F13, F14, F16, F19, F24, F29, F30, F33, F36, F37, F39, F41, F43, F46, F47
IoT Botnet	20	F1, F3, F6, F9, F11, F15, F17, F21, F23, F26, F27, F28, F31, F33, F38, F45, F49, F68, F75, F84

4.4. Accuracy and Loss Curve Analysis

The best fit is shown if the validation and training loss or accuracy increased and stabilized at a certain point (i.e., the model does not underfit or overfit). Both accuracy and loss are examined when the epoch size changes. Figure 5 illustrates how validation and training accuracy both rise linearly in IoT Botnet and then stabilize for a considerable amount of time beyond a certain threshold. It means that the model is perfectly fitted (no under- or overfit), and Figure 6 represents the training and validation loss of the ToN IoT dataset, which demonstrates that the validation and training losses will vary slightly. The model appears to fit.

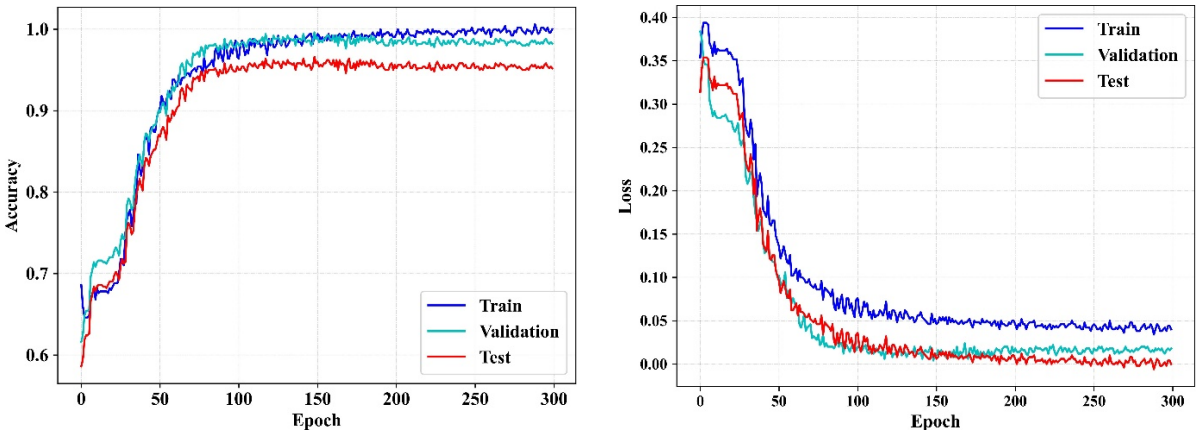

Figure 5. The loss and accuracy of IoT Botnet.

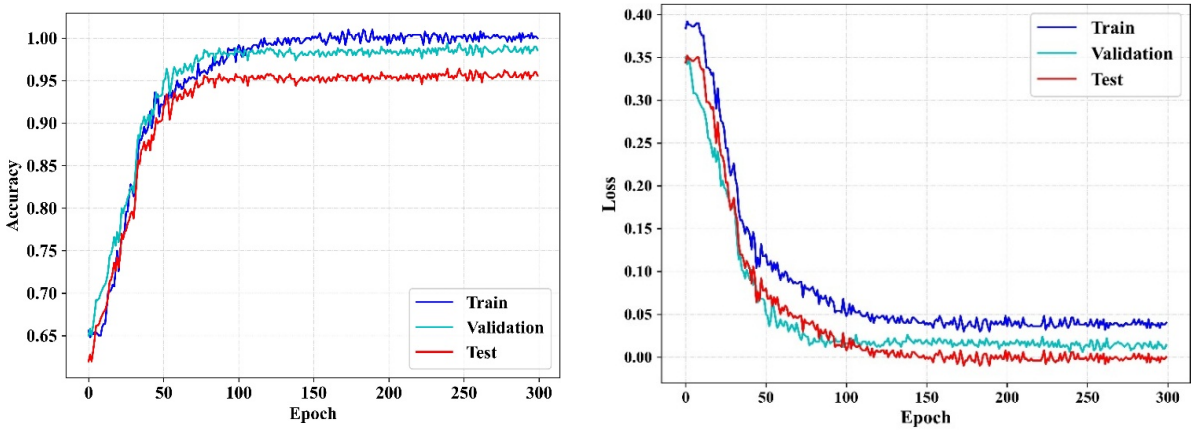

Figure 6. The loss and accuracy of ToN-IoT.

4.5. Confusion Matrix and ROC Analysis

The confusion matrix is the measurement that displays the classification's outcome. For example, it displays the accurate and inaccurate categorization results. For binary classification, it may have 2 × 2 dimensions. For multi-class classification, including N classes, it may have N × N dimensions. Despite not being a standalone statistic, the confusion matrix serves as a foundation for other measures that may be used to measure efficacy. Utilizing the confusion matrix (CM), the suggested attention-based GRU-based IDS's performance is examined. This provides a clear description of the classifier's behavior in identifying the samples included in the dataset's various classes. The classifying results of the attention-based GRU-based IDS employing the ToN-IoT and IoT Botnet datasets are displayed in Figure 7. In CM, the columns represent anticipated labels, while the rows represent genuine labels. The suggested attention-based GRU-based IDS employing the changed dataset obtains, according to Figure 7, an equivalent detection level to the original datasets.

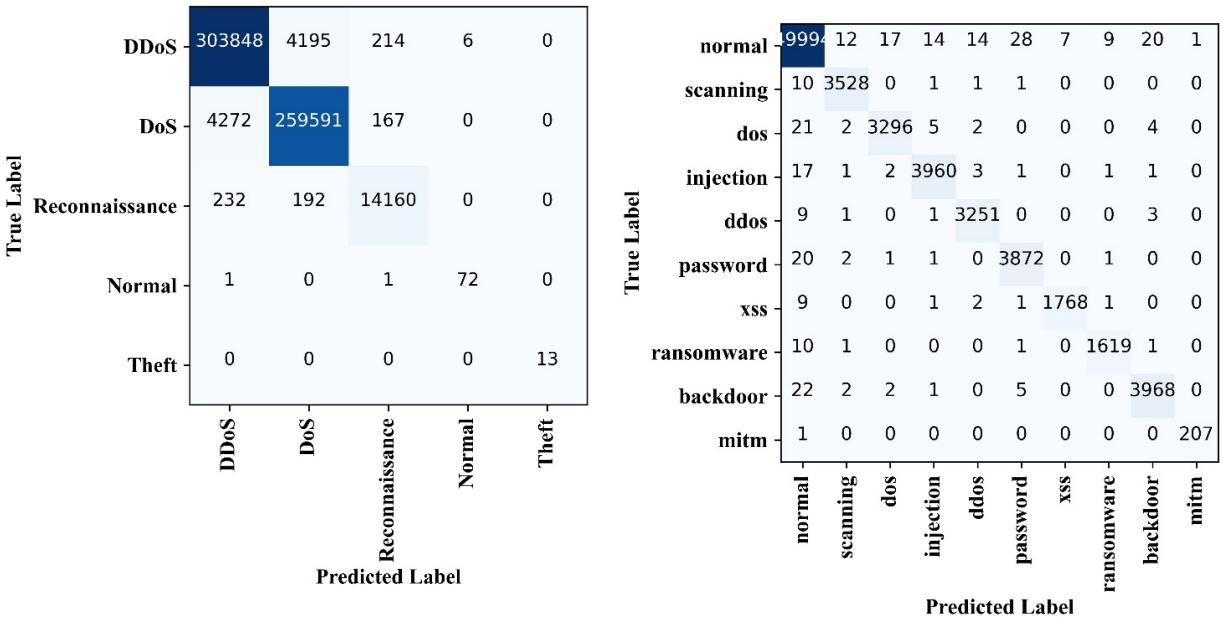

Figure 7. Confusion matrix of the PEDL using IoT Botnet and ToN-IoT datasets.

The ROC curve is a reliable measure for displaying the sensitivity and specificity linked to a continuous variable. It is a plot of coordinates with a vertical axis for the true

positive rate (TPR) and a horizontal axis for the false positive rate (FPR). The AUC, often known as the area under the ROC curve, is recognized as being a crucial evaluation metric.

$$ROC = \frac{p_f}{p_f + n_t} \tag{30}$$

The proposed model has better values than the existing models, as shown by the ROC curve which is represented in Figure 8.

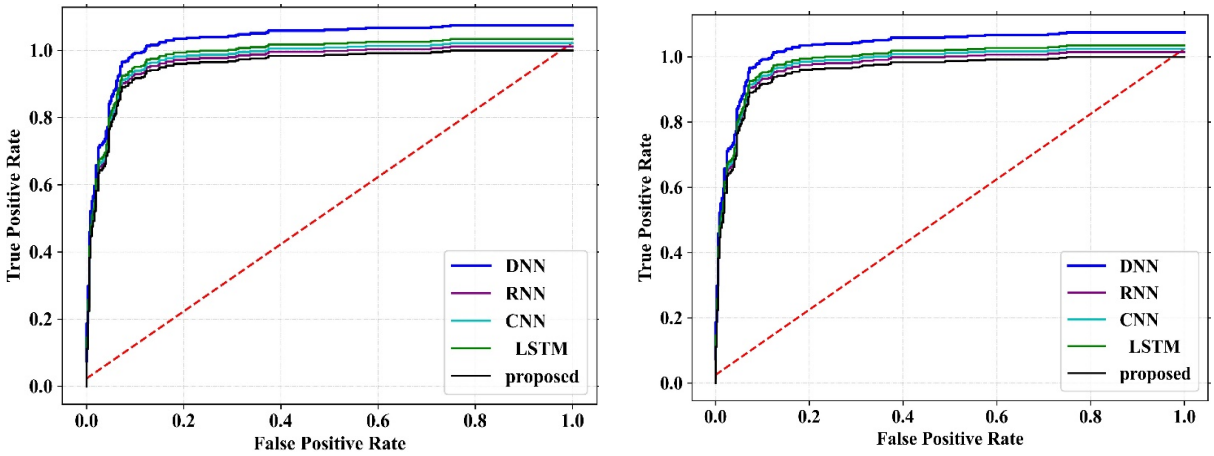

Figure 8. ROC curve of PEDL utilizing ToN-IoT and IoT Botnet datasets.

Some of the most effective evaluation metrics for imbalanced datasets are the class-wise prediction outcomes of accuracy, F1-score, and detection rate. The frequencies for the attack and normal classes are imbalanced in both datasets of IoT Botnet and ToN-IoT. Thus, data are calculated at the class level to assess the efficacy of the proposed attention-based GRU IDS. The initial class-wise prediction results and converted datasets are obtained using p_t, n_t, p_f, and n_f and depending on the DR, F1-score, and PR. The initial class-wise prediction results that were generated utilizing the ToN-IoT and IoT Botnet datasets are shown in Tables 6 and 7. It is important to note that the suggested attention-based GRU IDS has improved results for the majority of the threat vectors after implementing confidentiality and overall has a small difference compared to the original datasets.

Table 6. Class-wise prediction of ToN-IoT dataset.

TON-IOT Dataset				
Class	**Precision**	**Detection Rate**	**F1-Score**	**Accuracy**
normal	99.7%	99.7%	99.7%	99.6%
scanning	99.4%	99.2%	99.3%	99.9%
dos	99.1%	99.2%	99.1%	99.9%
injection	99.3%	99.3%	99.3%	99.9%
ddos	99.4%	99.4%	99.4%	99.9%
password	99.2%	99.3%	99.3%	99.9%
xss	99.2%	99.1%	99.2%	99.9%
ransomware	99%	99.3%	99.2%	99.9%
backdoor	99.3%	99.4%	99.4%	99.9%
mitm	99.5%	99.5%	99.5%	99.9%

Table 7. Class-wise prediction of ToN-IoT dataset.

IOT-Botnet Dataset				
Class	**Precision**	**Detection Rate**	**F1-Score**	**Accuracy**
DDoS	98.5%	98.5%	98.5%	98.5%
DoS	98.3%	98.3%	98.3%	98.5%
Reconnaissance	97.2%	97.1%	97.1%	99.8%
Normal	95.8%	94.5%	95.2%	99.9%
Theft	1	1	1	1

4.6. Comparisons with Existing Detection Models

Recent state-of-the-art methods were contrasted with the attention-based GRU IDS. For result comparison, DNN, RNN, CNN, and LSTM are four unique deep learning algorithms. The results of the existing techniques were acquired utilizing both of the datasets in the same situation. The performance of the attention-based GRU was assessed and compared using evaluation metrics, including class-level precision, detection rate, F1-score, and overall accuracy.

Figure 9a compares the proposed model's accuracy with existing models. The proposed model was achieved with an accuracy of 99.3%, which is high among the existing models because LSTM models require a large dataset to train it, so it becomes much slower than the other existing methods. DNN is a complex model with a large training time, but the accuracy is high compared to other existing models. The location and direction of an object are not encoded by CNN. There must be an absence of spatial invariance with respect to the supplied data and there must be lots of training data. The conventional RNN is too slow compared to the other existing methods, and it is also complex to train and faces problems such as gradient vanishing and exploding. Figure 9b shows the detection rate or recall. Here, the proposed technique attains a recall rate of 99.3%. The recall rate is higher than that of the other existing methods. The F1-score is displayed in Figure 9c. Here, the proposed technique attains 98.2%. The F1-score is higher than that of the other existing methods. Figure 9d shows the precision. The proposed technique attains precision of 98.2%. The precision is higher than that of the other existing methods.

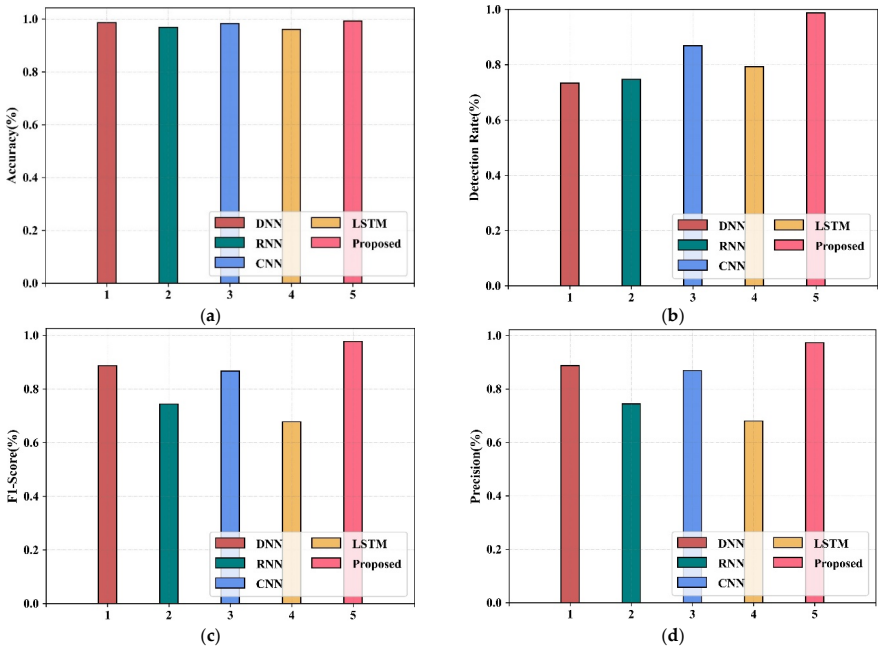

Figure 9. (**a**) Accuracy, (**b**) detection rate, (**c**) F1-score, and (**d**) precision of IoT Botnet dataset.

Figure 10a displays the accuracy of the proposed method. The accuracy is 99.8%. Compared to the other approaches now in use, the accuracy is higher. Figure 10b shows the detection rate or recall. The proposed method has a recall rate of 99.1%. The recall rate is higher than that of the other existing methods. The proposed model's F1-score is shown in Figure 10c. The recall rates for the suggested model are 99.1%. The F1-score is greater than that of the other existing techniques. The proposed model's precision is displayed in Figure 10d. The proposed method attains 99.1% accuracy. The precision is greater than that of the other existing methods.

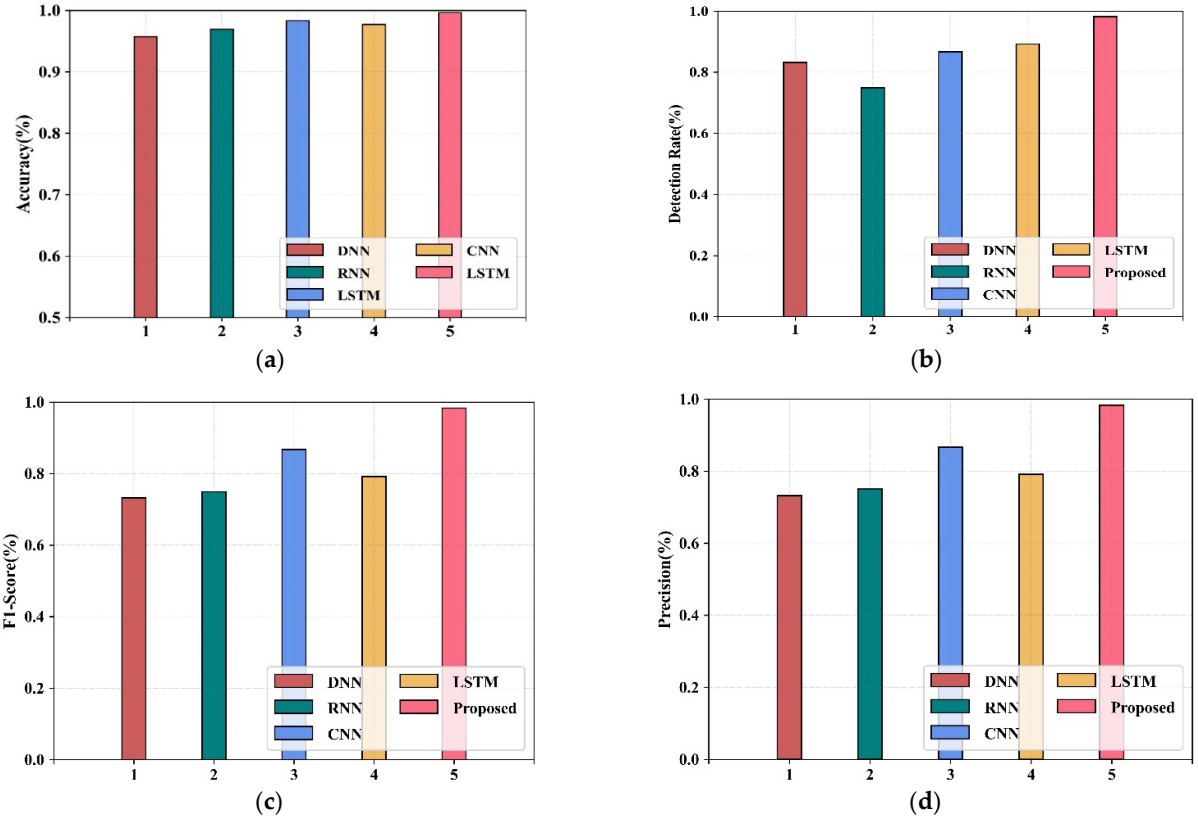

Figure 10. (**a**) Accuracy, (**b**) detection rate, (**c**) F1-score, and (**d**) precision of the ToN-IoT dataset.

In these figures, the proposed classifier obtained improved classification results as compared with other conventional deep learning models. Table 8 shows the performance comparison analysis over other exiting studies Kumar, S.D et al. [41].

Table 8. Comparison of proposed classification performance with other state-of-the-art studies.

Methods	F1-Score (%)	Recall (%)	Precision (%)
AGRU (Proposed)	98.2 for IoT Botnet, 99.1 for ToN-IoT	99.3 for IoT Botnet, 99.8 for ToN-IoT	98.2 for IoT Botnet, 99.1 for ToN-IoT
Generative Adversarial Networks (GAN)	94.82	95.55	92.37
Auto Encoder (AE)	72.38	75.26	69.48
MAD-GAN	87.54	89.23	85.41
Tad-GAN	93.48	94.13	91.08

By comparing it with other existing studies Vatti, R. et al [42], the detection performance of the proposed AGRU model is enhanced due to its higher efficiency. For each dataset, the proposed anomaly detection model generates improved results. The exist-

ing studies show several limitations while detecting the anomalies from the given input data, meaning that the detection outcomes are reduced Patel, B et al. [43]. Figure 11a–d shows the performance comparison of the proposed classification methods with other conventional RNNs.

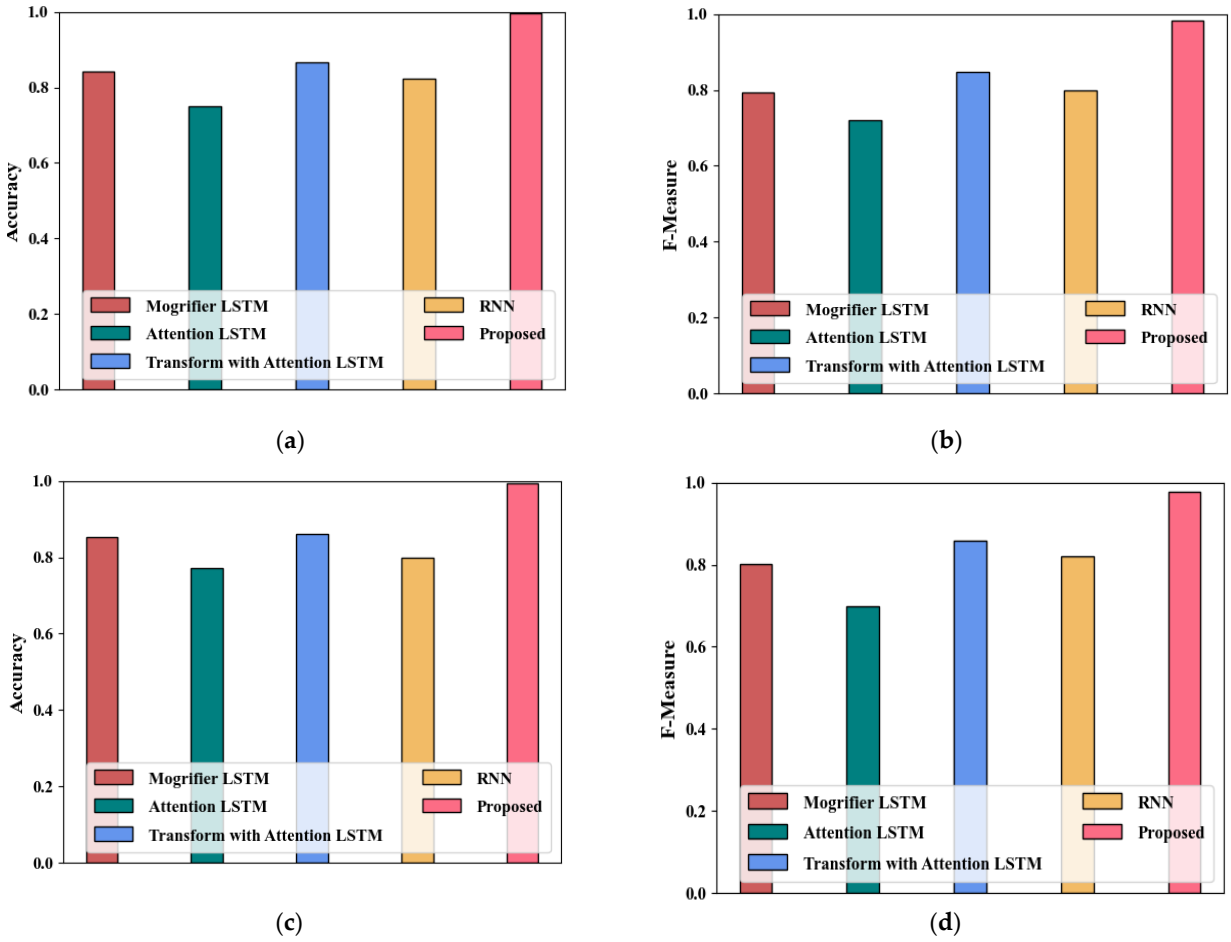

Figure 11. (**a**–**d**) Performance comparison of different RNNs with the proposed method. (**a**) Accuracy in IoT Botnet dataset. (**b**) F-measure in IoT Botnet dataset. (**c**) Accuracy in ToN-IoT dataset. (**d**) F-measure in ToN-IoT dataset.

Here, the proposed AGRU is analyzed using previous models, such as attention LSTM, Mogrifier LSTM, transform with attentional LSTM, and RNN. The analysis shows that the proposed detection model has improved performance compared to the existing models. The values attained for the comparison analysis with different RNNs are depicted in Table 9.

Table 9. Performance comparison of diverse RNNs with proposed AGRU.

Classification	Dataset-1		Dataset-2	
Methods	**Accuracy**	**F-Measure**	**Accuracy**	**F-Measure**
Proposed	0.996	0.983	0.993	0.977
Mogrifier LSTM	0.842	0.792	0.852	0.802
Attention LSTM	0.749	0.719	0.772	0.699
Transform with attention LSTM	0.867	0.847	0.861	0.857
RNN	0.822	0.8	0.8	0.82

On the other hand, the strength of the proposed SCAE model is analyzed by performing a comparison with other methods, and is illustrated in Figure 12.

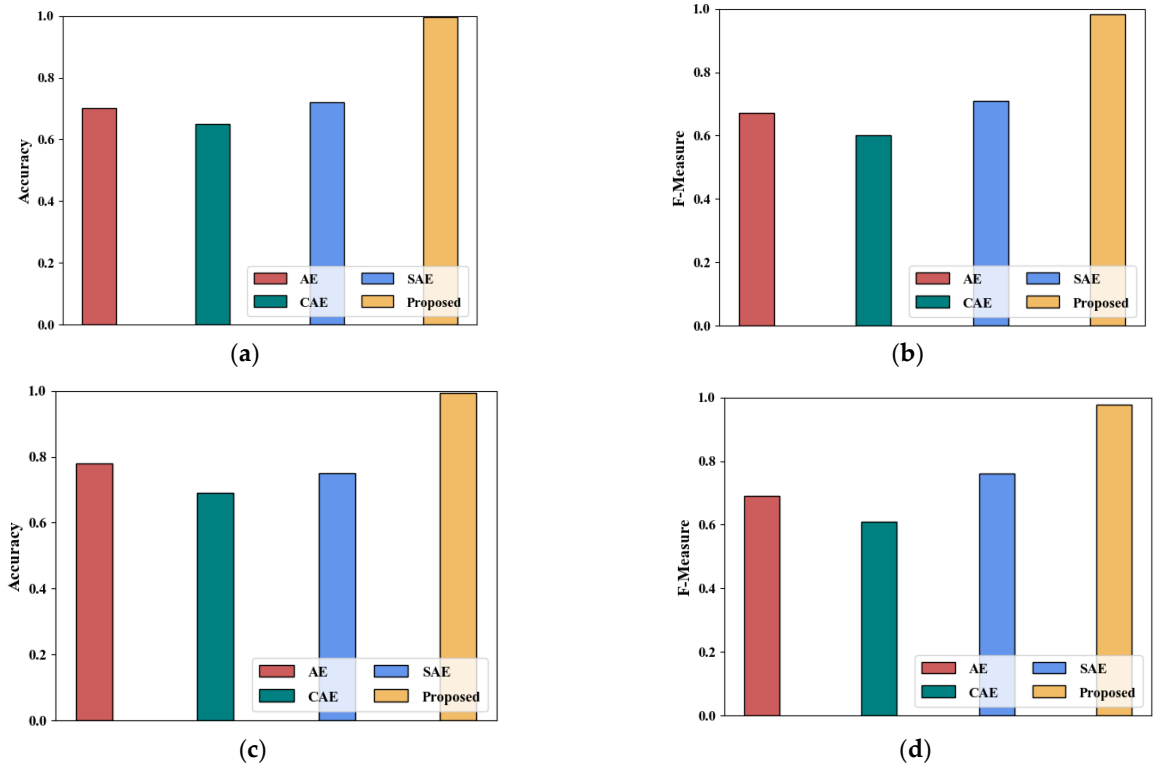

Figure 12. Performance analysis of proposed feature encoding model with other methods. (**a**) Accuracy in IoT Botnet dataset. (**b**) F-measure in IoT Botnet dataset. (**c**) Accuracy in ToN-IoT dataset. (**d**) F-measure in ToN-IoT dataset.

The above analysis proves the effectiveness of the proposed SCAE-based feature encoding model. Here, the efficiency of the proposed technique is compared to the existing models, such as auto encoder (AE), sparse auto encoder (SAE), and capsule auto encoder (CAE). The result analysis proves that the proposed model achieved improved performance compared to the other methods. The values obtained for the feature encoding comparison are portrayed in Table 10.

Table 10. Comparison analysis of proposed feature encoding method over others.

Methods	Dataset 1		Dataset 2	
Proposed	0.996	0.983	0.993	0.977
AE	0.702	0.672	0.78	0.69
CAE	0.65	0.6	0.69	0.61
SAE	0.72	0.71	0.75	0.76

The above results show the efficacy of the proposed methods and they show that the proposed techniques are highly suitable for detecting anomalies in IoT devices for the smart agriculture domain.

Figure 13 shows the time complexity analysis. Here, the proposed AGRU design's CPU operation time and memory usage linearly increase. Here, the time complexity of the proposed AGRU is lower than those of the existing LSTM, DNN, CNN, and RNN designs.

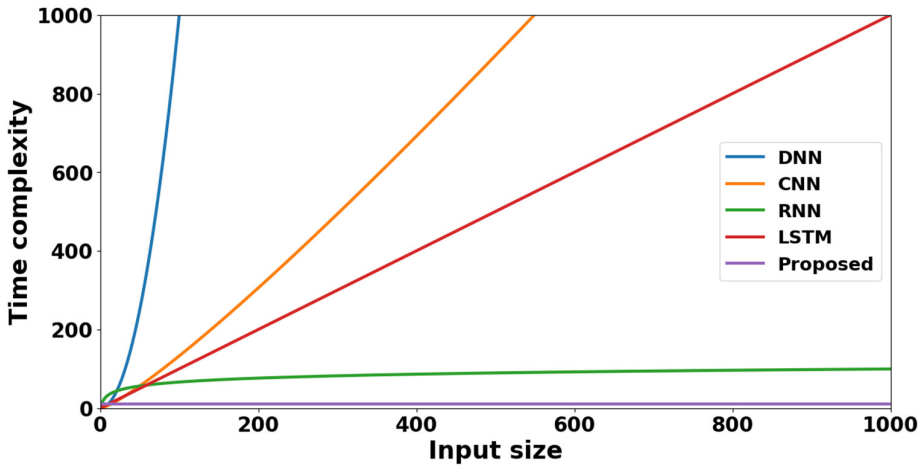

Figure 13. Performance analysis of time complexity.

5. Conclusions

Attacks and vulnerabilities in smart farming demonstrate the significance of security in IoT-based farming. With the help of encryption and intrusion detection using an attention-based gated recurrent unit neural network algorithm and sparse capsule auto encoder, we suggested a privacy-encoding-based enhanced deep learning framework. We examined how to construct a reliable encryption and intrusion detection scheme for an IoT network environment. To prevent deduction attacks, SCAE was used to change information into a new encrypted format. By enhancing a GRU, we put forward a neural network model called AGRU. Then, we suggested a fresh approach to intrusion detection using the AGRU. This approach might significantly increase intrusion detection's potency. We tested the effectiveness of our proposed AGRU-based intrusion detection technique using simulated trials. The experiment results demonstrated that our suggested AGRU-based intrusion detection approach could produce a significant improvement in effectiveness and accuracy when compared to several current intrusion detection methods including LSTM, CNN, RNN, and DNN. The proposed model has 99.9% accuracy, 99.7% precision, 99.9% recall, and 99.8% F1-score, respectively. In this proposed work, the hyper parameters were not optimally tuned. By considering this limitation, an efficient optimization algorithm will be introduced in future work for a parameter tuning purpose. We intend to carry out more research on intrusion detection and develop an improved encryption method for smart agriculture in the future. Additionally, we will focus on the categorization and early detection time analysis of more cyber-malware in complex systems in the actual world.

Author Contributions: K.K.: conceptualization, data curation, formal analysis, investigation, resources, software, and writing—original draft. P.G.: methodology, project administration, supervision, validation, visualization, writing—review and editing, and funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Two of the latest IoT-based publicly accessible datasets were utilized to confirm the efficacy of the suggested PEDL approach. The training and test sets were divided into 70% and 30%, respectively. The data that helped the findings of this study are openly available at the following URL/DOI: https://research.unsw.edu.au/projects/bot-iot-dataset; https://ieee-dataport.org/documents/toniot-datasets(accessed on 16 April 2023).

Acknowledgments: We declare that this manuscript is original, has not been published before, and is not currently being considered for publication elsewhere.

Conflicts of Interest: The authors affirm that they have no known financial or interpersonal conflicts that would have appeared to have an impact on the research presented in this paper.

References

1. Rezk, N.G.; Hemdan, E.E.D.; Attia, A.F.; El-Sayed, A.; El-Rashidy, M.A. An efficient IoT based smart farming system using machine learning algorithms. *Multimed. Tools Appl.* **2021**, *80*, 773–797. [CrossRef]
2. Varghese, R.; Sharma, S. Affordable smart farming using IoT and machine learning. In Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 14–15 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 645–650.
3. Popa, M.; Prostean, O.; Popa, A.S. Machine Learning Approach for Agricultural IoT in Proc. *Int. J. Recent Technol. Eng.* **2019**, 22–29.
4. Vij, A.; Vijendra, S.; Jain, A.; Bajaj, S.; Bassi, A.; Sharma, A. IoT and machine learning approaches for automation of farm irrigation system. *Procedia Comput. Sci.* **2020**, *167*, 1250–1257. [CrossRef]
5. Ouafiq, E.M.; Saadane, R.; Chehri, A. Data Management and Integration of Low Power Consumption Embedded Devices IoT for Transforming Smart Agriculture into Actionable Knowledge. *Agriculture* **2022**, *12*, 329. [CrossRef]
6. Reddy, K.S.P.; Roopa, Y.M.; LN, K.R.; Nandan, N.S. IoT based smart agriculture using machine learning. In Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 15–17 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 130–134.
7. Maduranga, M.W.P.; Abeysekera, R. Machine learning applications in IoT based agriculture and smart farming: A review. *Int. J. Eng. Appl. Sci. Technol.* **2020**, *4*, 24–27. [CrossRef]
8. Muniasamy, A. Machine learning for smart farming: A focus on desert agriculture. In Proceedings of the 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 9–10 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
9. Rehman, A.; Liu, J.; Keqiu, L.; Mateen, A.; Yasin, M.Q. Machine learning prediction analysis using IoT for smart farming. *Int. J.* **2020**, *8*, 6482–6487.
10. Xu, D. Agricultural climate change based on remote sensing image and emergency material supply management of agriculture, rural areas and farmers. *Arab. J. Geosci.* **2021**, *14*, 894. [CrossRef]
11. Bhanu, K.N.; Jasmine, H.J.; Mahadevaswamy, H.S. Machine learning implementation in IoT based intelligent system for agriculture. In Proceedings of the 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 5–7 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
12. Sarangdhar, A.A.; Pawar, V.R. Machine learning regression technique for cotton leaf disease detection and controlling using IoT. In Proceedings of the 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 20–22 April 2017; IEEE: Piscataway, NJ, USA, 2017; Volume 2, pp. 449–454.
13. Ramesh, S.; Vydeki, D. Rice blast disease detection and classification using machine learning algorithm. In Proceedings of the 2018 2nd International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), Ghaziabad, India, 20–21 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 255–259.
14. Kundu, N.; Rani, G.; Dhaka, V.S.; Gupta, K.; Nayak, S.C.; Verma, S.; Ijaz, M.F.; Woźniak, M. IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* **2021**, *21*, 5386. [CrossRef]
15. Jaisakthi, S.M.; Mirunalini, P.; Thenmozhi, D. Grape leaf disease identification using machine learning techniques. In Proceedings of the 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 21–23 February 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
16. Rumy, S.S.H.; Hossain, M.I.A.; Jahan, F.; Tanvin, T. An IoT based System with Edge Intelligence for Rice Leaf Disease Detection using Machine Learning. In Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 21–24 April 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
17. Nawaz, M.A.; Rasool, R.M.; Kausar, M.; Usman, A.; Bukht, T.F.N.; Ahmad, R.; Jaleel, A. Plant disease detection using internet of thing (IoT). *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 505–509. [CrossRef]
18. Garg, S.; Pundir, P.; Jindal, H.; Saini, H.; Garg, S. Towards a multimodal system for precision agriculture using IoT and machine learning. In Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–7.
19. Vasavi, P.; Punitha, A.; Rao, T.V.N. Crop leaf disease detection and classification using machine learning and deep learning algorithms by visual symptoms: A review. *Int. J. Electr. Comput. Eng.* **2022**, *12*, 2079. [CrossRef]
20. Aggarwal, S.; Suchithra, M.; Chandramouli, N.; Sarada, M.; Verma, A.; Vetrithangam, D.; Pant, B.; Ambachew Adugna, B. Rice Disease Detection Using Artificial Intelligence and Machine Learning Techniques to Improvise Agro-Business. *Sci. Program.* **2022**, *2022*, 1757888. [CrossRef]
21. Visconti, P.; Giannoccaro, N.I.; de Fazio, R.; Strazzella, S.; Cafagna, D. IoT-oriented software platform applied to sensors-based farming facility with smartphone farmer app. *Bull. Electr. Eng. Inform.* **2020**, *9*, 1095–1105. [CrossRef]
22. Anand, R.; Sethi, D.; Sharma, K.; Gambhir, P. Soil moisture and atmosphere components detection system using IoT and machine learning. In Proceedings of the 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 27–29 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 842–847.
23. Vincent, D.R.; Deepa, N.; Elavarasan, D.; Srinivasan, K.; Chauhdary, S.H.; Iwendi, C. Sensors driven AI-based agriculture recommendation model for assessing land suitability. *Sensors* **2019**, *19*, 3667. [CrossRef]

24. Mahmoudzadeh, H.; Matinfar, H.R.; Taghizadeh-Mehrjardi, R.; Kerry, R. Spatial prediction of soil organic carbon using machine learning techniques in western Iran. *Geoderma Reg.* **2020**, *21*, e00260. [CrossRef]
25. Angin, P.; Anisi, M.H.; Göksel, F.; Gürsoy, C.; Büyükgülcü, A. AgriLoRa: A digital twin framework for smart agriculture. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.* **2020**, *11*, 77–96.
26. Ali, M.; Kanwal, N.; Hussain, A.; Samiullah, F.; Iftikhar, A.; Qamar, M. IoT based smart garden monitoring system using NodeMCU microcontroller. *Int. J. Adv. Appl. Sci.* **2020**, *7*, 117–124.
27. Ali, R.; Pal, A.K.; Kumari, S.; Karuppiah, M.; Conti, M. A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. *Future Gener. Comput. Syst.* **2018**, *84*, 200–215. [CrossRef]
28. Bothe, A.; Bauer, J.; Aschenbruck, N. RFID-assisted continuous user authentication for IoT-based smart farming. In Proceedings of the 2019 IEEE international conference on RFID technology and applications (RFID-TA), Pisa, Italy, 25–27 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 505–510.
29. Di Mauro, M.; Galatro, G.; Liotta, A. Experimental review of neural-based approaches for network intrusion management. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2480–2495. [CrossRef]
30. Dong, S.; Xia, Y.; Peng, T. Network abnormal traffic detection model based on semi-supervised deep reinforcement learning. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 4197–4212. [CrossRef]
31. Pelletier, C.; Webb, G.I.; Petitjean, F. Deep learning for the classification of Sentinel-2 image time series. In Proceedings of the IGARSS 2019–2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 461–464.
32. Rangwani, D.; Sadhukhan, D.; Ray, S.; Khan, M.K.; Dasgupta, M. An improved privacy preserving remote user authentication scheme for agricultural wireless sensor network. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4218. [CrossRef]
33. Vangala, A.; Das, A.K.; Lee, J.H. Provably secure signature-based anonymous user authentication protocol in an Internet of Things-enabled intelligent precision agricultural environment. *Concurr. Comput. Pract. Exp.* **2021**, e6187. [CrossRef]
34. Vidyashree, L.; Suresha, B.M. Methodology to secure agricultural data in IoT. In *Emerging Technologies in Data Mining and Information Security*; Springer: Singapore, 2019; pp. 129–139.
35. Chen, M.; Lee, T.F.; Pan, J.I. An enhanced lightweight dynamic pseudonym identity based authentication and key agreement scheme using wireless sensor networks for agriculture monitoring. *Sensors* **2019**, *19*, 1146. [CrossRef]
36. Gupta, A.; Nahar, P. Classification and yield prediction in smart agriculture system using IoT. *J. Ambient. Intell. Humaniz. Comput.* **2022**, 1–10. [CrossRef]
37. Bakthavatchalam, K.; Karthik, B.; Thiruvengadam, V.; Muthal, S.; Jose, D.; Kotecha, K.; Varadarajan, V. IoT framework for measurement and precision agriculture: Predicting the crop using machine learning algorithms. *Technologies* **2022**, *10*, 13. [CrossRef]
38. Colombo-Mendoza, L.O.; Paredes-Valverde, M.A.; Salas-Zárate, M.D.P.; Valencia-García, R. Internet of Things-driven data mining for smart crop production prediction in the peasant farming domain. *Appl. Sci.* **2022**, *12*, 1940. [CrossRef]
39. Murugamani, C.; Shitharth, S.; Hemalatha, S.; Kshirsagar, P.R.; Riyazuddin, K.; Naveed, Q.N.; Islam, S.; Ali, S.P.M.; Batu, A. Machine Learning Technique for Precision Agriculture Applications in 5G-Based Internet of Things. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 6534238. [CrossRef]
40. Raghuvanshi, A.; Singh, U.K.; Sajja, G.S.; Pallathadka, H.; Asenso, E.; Kamal, M.; Singh, A.; Phasinam, K. Intrusion detection using machine learning for risk mitigation in IoT-enabled smart irrigation in smart farming. *J. Food Qual.* **2022**, *2022*, 3955514. [CrossRef]
41. Kumar, S.D.; Esakkirajan, S.; Bama, S.; Keerthiveena, B. A microcontroller based machine vision approach for tomato grading and sorting using SVM classifier. *Microprocess. Microsyst.* **2020**, *76*, 103090.
42. Vatti, R.; Vatti, N.; Mahender, K.; Vatti, P.L.; Krishnaveni, B. Solar energy harvesting for smart farming using nanomaterial and machine learning. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Tokyo, Japan, 2020; Volume 981, p. 032009.
43. Patel, B.; Sharaff, A. Rice crop disease prediction using machine learning technique. *Int. J. Agric. Environ. Inf. Syst.* **2021**, *12*, 1–15. [CrossRef]