Acadlore Transactions on AI and Machine Learning
https://www.acadlore.com/journals/ATAIML

Detecting False Data Injection Attacks in Industrial Internet of Things Using an Optimized Bidirectional Gated Recurrent Unit-Swarm Optimization Algorithm Model

Nadella Sree Divya[1*], Ramesh Vatambeti[2]

[1] Department of Information Technology, Mahatma Gandhi Institute of Technology, 500075 Hyderabad, India
[2] School of Computer Science and Engineering, VIT-AP University, 522237 Vijayawada, India

* Correspondence: Ramesh Vatambeti (ramesh.v@vitap.ac.in)

Received: 03-20-2023 **Revised:** 04-10-2023 **Accepted:** 04-28-2023

Citation: N. S. Divya and R. Vatambeti, "Detecting false data injection attacks in industrial Internet of Things using an optimized bidirectional gated recurrent unit-swarm optimization algorithm model," *Acadlore Trans. Mach. Learn.*, vol. 2, no. 2, pp. 75–83, 2023. https://doi.org/10.56578/ataiml020203.

Abstract: The rapid adoption of the Industrial Internet of Things (IIoT) paradigm has left systems vulnerable due to insufficient security measures. False data injection attacks (FDIAs) present a significant security concern in IIoT, as they aim to deceive industrial platforms by manipulating sensor readings. Traditional threat detection methods have proven inadequate in addressing FDIAs, and most existing countermeasures overlook the necessity of validating data, particularly in the context of data clustering services. To address this issue, this study proposes an innovative approach for FDIA detection using an optimized bidirectional gated recurrent unit (BiGRU) model, with the Sailfish Optimization Algorithm (SOA) employed to select optimal weights. The proposed model exploits temporal and spatial correlations in sensor data to identify fabricated information and subsequently cleanse the affected data. Evaluation results demonstrate the effectiveness of the proposed method in detecting FDIAs, outperforming state-of-the-art techniques in the same task. Furthermore, the data cleaning process showcased the ability to recover damaged or corrupted data, providing an additional advantage.

Keywords: Industrial Internet of Things; Data clustering; False data injection attack; Bidirectional gated recurrent unit; Sailfish optimization algorithm; Machine learning

1 Introduction

Internet of Things (IoT) is deeply ingrained in many dense and complicated domains, such as industrial operations, logistics, public security, and smart cities [1, 2]. It is difficult to study the smart industry because of the complex interconnectedness and heterogeneity of its many sub-domains. In particular, the IIoT paradigm is notable for the ease of connecting and coordinating industrial equipment with one another. In addition, the IIoT is considered as a component of Industry 4.0 or the 4th Industrial Revolution [3], since the cyber physical system (CPS) provides the connection between the physical and digital environments. For example, the integration of logistics with the production line feeds information to the control centre [4, 5], and autonomous robots have been used to manage production in a fully linked factory. By having data at fingertips, manufacturing time is cut down and diagnoses are offered in real time, boosting efficiency all around. Hence, IIoT is crucial for gathering, sharing, and managing the quantity of data needed by various applications for making decisions [6]. IIoT services collect the huge amounts of information they have produced from a wide range of sources, and then share the information and make it easy to be accessed safely [7]. Thus, the industrial applicability and reliability expansion of the IIoT depends critically on the security of its data dissemination service [8]. These applications rely on data to make decisions. If the data are inconsistent or manipulated in any way, the complete industrial system may display unexpected outcomes. Consequently, the IIoT is vulnerable to a variety of security threats [9] due to the challenges posed by managing and distributing the massive amounts of data generated by interactions between many devices. Due to data attacker's unpredictability, the FDIA stands out as one of the most destructive intrusion networks in the IIoT [10]. As hostile devices authorized in the networks carry out their usual data gathering tasks, detecting a FDIA becomes a non-trivial effort due to the attack's complexity. Nevertheless, the attack might occur at any time, causing chaos in the network [11]. The FDIA occurs either when an outsider takes control of the device and modifies the data or when the device

itself acts inappropriately and modifies the data, which makes it more challenging to detect the attack and lengthens the network downtime [12, 13]. In addition, there are a number of additional attacks, such as concealment, suggestion, and deduction, which function similarly to the FDIA but have their own quirks. As a result, when rogue devices in the IIoT are quickly identified and isolated, extended durations of network failures, which might damage the data dissemination service, are reduced, thus reducing the likelihood that the misbehaving devices continue to create inconsistent sensed data [14]. The most popular methods include en-route filtering, detection, machine learning (ML), and identification detection system (IDS). Most Wireless Sensor Networks (WSNs) use en route filtering schemes, which rely on intermediate nodes' reports on the integrity of packets en route to their final destinations [15]. The reports are validated to ensure their veracity, but they don't account for any shifts in the detected data. As an alternate method of countering FDIAs, smart grids have used collaborative detection systems in which each device serves as a detection agent. Training machines use ML techniques in general to detect many attacks simultaneously and supply defensive measures tailored to each threat [16]. Finally, intrusion detection systems (IDSs) deal with attacks using various systems in many situations and monitor them, which might lead to excessive resource usage and fresh security holes. Hence, solutions that identify and isolate risks actively are required for the development of IIoT, thus guaranteeing better resilience of the data gathering and dissemination services. In this study, BiGRU was implemented as a classification strategy for spotting malicious data injection attacks. Unsupervised learning allowed the model to uncover latent correlation patterns in the data, which then may be used to detect corrupted data by measuring how differently it behaves from the "learned" correlation structure. The autoencoder, proposed in this study, learned correlation through two dimensions of time and space, which might reflect hidden layers of the correlation model more accurately. The rest of the study was organized as follows: the technical context was provided in Section II; the proposed system was described in detail in Section III; effectiveness of the proposed model was assessed in Section IV.

2 Literature Review

Wei et al. [17] offered a forecasting-aided state estimation (FASE) method, which provided more accurate estimations using an unscented Kalman filter (UKF). The projection statistics may be used to successfully detect and suppress random outliers by transforming the filtering step of the projection statistics (PS). The gathered state estimation (SE) findings were then used to develop a generalized likelihood ratio test (GLRT) for identifying FDIAs across successive snapshots. The GLRT used a low false alarm rate to evaluate the distance between two innovation sequences using the Dynamic Time Warping (DTW) method. Extensive numerical simulations verified the viability of the projected FDIA and the efficacy of the proposed detection approach. Vincent et al. [18] presented a graph convolutional network (GCN) framework to find FDIAs. This method observed FDIAs graphically, analyzed the changing state estimate values based on the scheme architecture and pinpointing the exact position of any FDIA. Then the method was applied to the 2848-bus systems to measure its performance. In computer simulations, the suggested method effectively identified FDIAs in both small and large systems with reasonable accuracy and detection time when taking into account varying disturbance magnitudes and attack sparsity. In addition to comparing Distribution System State Estimation (DSSE) findings with the Weighted Least Squares (WLS) algorithm, a standard model-based method, Radhoush et al. [19] offered an alternative to the conventional methodology of detecting FDIAs and performing SE calculations independently. In the case of inaccurate measurements, the DSSE performance of the projected technique was superior to that of the WLS method and the independent DSSE/FDIA method, and the suggested method also ran more quickly. Two case studies, with one utilizing a modified distribution scheme with distributed generations (DG), were used to verify the efficacy of the proposed technique using two FDIA methods. The outcomes demonstrated that the suggested strategy outperformed binary classification alone in terms of accuracy and F1-score. For every phasor measurement unit (PMU) measurement, the suggested approach correctly identified the FDIAs. In addition, the suggested strategy outperformed the regression-only and WLS approaches when dealing with imperfect data, as measured by the DSSE. Miao et al. [20] created a nonlinear CPS to solve the problem of adaptive security control against FDIAs on the sensor and actuator. The feedback control scheme could not reach the classical error surface because the sensor and actuator were destroyed. The first step against sensor attacks was building a state observer. Then the nonlinear term was approximated using neural networks, which dampened state-dependent actuator attacks. In addition, a unique time-varying symmetry barrier function design was constructed that may accomplish individualized output signal limitations while protecting against FDIAs. The aforementioned control technique provided a solution to the output problem faced by CPSs when subjected to FDIAs. Finally, an example of numerical simulation was provided to show how the suggested controller worked. Hua and Hao [21] studied the security issue in CPS by using a multi-sensory paradigm. Each sensor sent data in packets to a distant estimator across wireless channels, where spoofed information might be introduced. To address this, a modified multi-sensor Kalman filter fusion technique was presented, based on the information from trustworthy sensors and the correlation between trustworthy and untrustworthy sensors. With the help of the suggested technique, a generalized linear FDIA approach was proposed, with a Gaussian distribution centred on any arbitrary mean. An anticipated State-

Action-Reward-State-Action (SARSA)-based attack detection method was developed to further enhance the detection performance. Finally, Unmanned Aerial Vehicle (UAV)-based simulation results were presented to demonstrate the practicality and efficiency of the solutions.

3 Proposed System

This section described the existing issue, the accepted model, the types of attacks, and the recommended detection method.

3.1 System Setup

Hydraulic system was used as an example in this study. In order to track the status of the hydraulic system, a dataset of a small amount of sensor signals was collected, which was not a simulation result but rather the actual hydraulic testing result. The system quantitatively varied the state of four hydraulic components by cyclical repetition of constant load cycles and measurement of process parameters. The dataset included data of temperature, pressure, flow rate, and power of all 15 sensors, as well as raw process sensor data (i.e., without feature extraction). About 132,100 to 132,300 data points were collected from each sensor. The fewest possible measurements were used in order to prevent useless input vectors.

3.2 FDIA Description

This study intended to uncover FDIA. Based on the attack model given in the study of Wang et al. [22], application of the method of this study was tailored to the requirements and characteristics of IIoT systems. In this paradigm, an attacker tampered with and/or falsely injected data from a single sensor or several sensors at any moment, and the tampered data still fell within the acceptable range of legitimate measurements. An attack on the integrity of measured data was defined. Let z_a be a set of potentially inaccurate values. Due to $z = [z1, \ldots, zm]$, z_a was written as $z_a = z + a$. Let T be clean measurements, $a = [a1, \ldots, am]$, and T be a new data vector created. As the attacker faked the i-th reading and replaced the corresponding authentic value with a false value, the vector a represented the attack vector when a_i, which is the i-th element of a, was nonzero.

3.3 Proposed Attack Detection Algorithm

The BiGRU model was used to categorize the attack. Weights were updated in primary deep learning (DL) networks and their derivative models like convolutional neural network (CNN) throughout the backpropagation process, which led to problems of inflating and disappearing gradients. Recurrent neural networks (RNN), such as Long Short-Term Memory (LSTM) and gated recurrent unit (GRU), were developed to address these problems. In contrast to GRU, LSTM was not a good and acceptable model because it had large temporal complexity and many parameters. Compared with more traditional DL techniques, such as Multi-Layer Perceptron (MLP), CNN, and LSTM, the GRU proved to be a superior classification method. In addition, GRU extracted useful characteristics to identify energy theft. Update and reset gates were available on the GRU. On the one hand, the GRU algorithm relied on the former gate to determine what information from the past should be forwarded. On the other hand, the model made use of the latter gate to decide how much of the past evidence to discard. Eqns. (1)-(4) provide full calculations for the update gate, reset gate, potential hidden state, and final hidden state.

$$v_t = \{\, sigmoid * (W_v x_t + W_v h_{t-1} + b_v)\} \tag{1}$$

$$r_t = \{\, sigmoid * (W_r x_t + W_r h_{t-1} + b_r)\} \tag{2}$$

$$h'_t = \{\tanh * (W_{x_t} + W_{r_t} \odot h_{t-1})\} \tag{3}$$

$$h_t = \{v_t \odot h_{t-1} + (1 - v_t) \odot h'_t\} \tag{4}$$

where, v_t is the update gate, r_t is the reset gate, h'_t is the applicant hidden state, h_t is the new hidden state, W is the weight, h_{t-1} is the hidden State, b is the bias, $tanh$ is the hyperbolic activation function, and the icon is the Hadamard product. Compared with unidirectional models, the bidirectional ones used both past and future data to make predictions about the present. BiGRU was improved based on the original GRU. Apart from optics, telecommunications, and computer networking, Natural Language Processing (NLP) was also used in language study and detection of structural damage. Unfortunately, it had limited value outside of engineering, particularly the detection of attacks. As a result, the BiGRU was used in Stochastic gradient (SG) in this study. Figure 1 depicts the BiGRU generic architecture. BiGRU was the grouping of two unidirectional GRUs in opposite directions. The first GRU, designated as GRU 1 (sometimes referred to as "forward GRU"), moved from the left to the right,

whereas the second GRU, designated as GRU 2 (sometimes referred to as "backward GRU"), moved in the opposite direction. Based on the past knowledge and future situations, the final prediction of the present facts and observations necessitated a left and back movement again. The incoming data sequence was first processed using backward and forward neural networks, which was the core operating mechanism of BiGRU. Finally, the two sets of data were mixed in the final output layer. The five-layer model for theft categorization used two BiGRU layers, a flatten layer, a dropout layer, and a dense layer. Both BiGRU layers had 50 neurons, and the dropout probability of the dropout layer was set at 0.2. A single neuron with a sigmoid activation function was used in the dense layer to finally get the desired output. In addition, BiGRU learned and remembered the long-term temporal connections between the traits through its memory. Mathematical modelling of the BiGRU model was accomplished through Eqns. (5)-(7) [23].

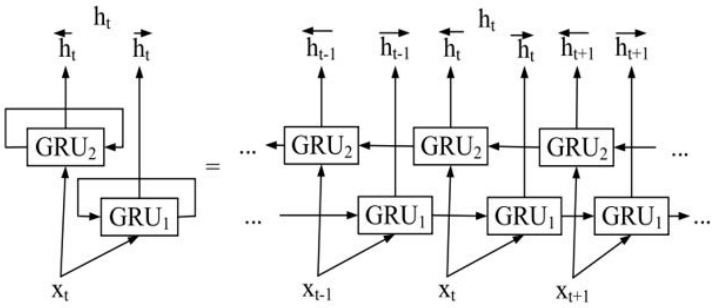

Figure 1. Generic framework of BiGRU

$$\vec{h}_t = GRU_1\left(x_t, \overrightarrow{h_{t-1}}\right) \tag{5}$$

$$\overleftarrow{h_t} = GRU_2\left(x_t, \overleftarrow{h_{t+1}}\right) \tag{6}$$

$$h_{ut} = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{7}$$

where, $\vec{h}_{ut}, \vec{h}_t, \overrightarrow{h_{t-1}}$ and $\overleftarrow{h_{t+1}}$ are the current and revised GRU 1 and GRU 2 states. Two vectors were joined together with this symbol. Sailfish optimization, as explained below, was used to choose the ideal weight of the BiGRU.

3.3.1 SOA

The sailfish algorithm [24] simulated the behaviour of a school of sailfish as they seek prey in waves. The sardine and sailfish inhabitants were seeded first with a random number generator. Second, the sailfish employed alternating attacks to weaken the sardines' group defense. Finally, the sailfish completed the position optimization by searching and capturing appropriate sardines. 1) Sailfish location update Sailfish used the attack replacement tactic during hunting, and learned to work together during attacks by coordinating their movements. This was the updated position formula:

$$X^i_{\text{newSF}} = X^i_{\text{eliteSF}} - \lambda_i \times (\text{rand}(0,1)) \times \left(\frac{X^i_{\text{eliteSF}} + X^i_{\text{injureds}}}{2}\right) - X^i_{\text{oldSF}} \tag{8}$$

where, X^i_{oldSF} is the current location of the sailfish, X_newSF^i is Sailfish's current locations have been updated, i is the most fit elite sailfish, $X^i injuredS$ is the most fit damaged sardine, and rand(0,1) is a random sum between 0 and 1. In the following formula, i is the iteration coefficient of the i-th iteration, and PD is the prey density: There was:

$$PD = 1 - \left(\frac{N_{SF}}{N_{SF} + N_S}\right) \tag{9}$$

where, N_{SF} and N_S are the statistics. The number of sailfish was expressed as:

$$N_{SF} = N * \text{ Percent} \tag{10}$$

where, N is the total number of sardine inhabitants, and $Percent$ is the share of total number. 2) Sardine location update When the sardine was observed, its position information formula was:

$$X^i_{\text{newS}} = r * \left(X^i_{\text{eliteSF}} - X^i_{\text{oldS}} + AP\right) \tag{11}$$

where, X^i_{olds} is the current position of the sardine, X^i_{news} is the new position that the sardine has moved to recently, AP is the best attack power of sailfish, and r is a random value between 0 and 1. The attack power of sailfish decreased as the iteration count increased as follows:

$$AP = A * (1 - 2 * t * e) \tag{12}$$

where, t is the current iteration number, A is the attack power conversion control, and e is the energy efficiency. Convergence was hastened by reducing the sailfish's attack strength, which varied linearly from A to 0. All positions of sardine should be updated when AP was greater than 0.5, and their partial positions should be updated when AP was less than 0.5. Their positions were shown as follows:

$$a = N_S * AP \tag{13}$$

$$\beta = d_i * AP \tag{14}$$

where, d_i is the number of dimensions, d_i the number of independent variables, and d is the number of items to be updated. 3) Predation stage In the last hunting phase, the sailfish was updated to the position of any sardine it had killed, including any sardine hurt during an earlier iteration. The following equation was used to adjust the position:

$$X^i_{SF} = X^i_S \text{ if } f\left(S_i\right) < f\left(SF_i\right) \tag{15}$$

where, X^i_{SF} and X^i_S are the current positions of sailfish and sardine, and $f\left(SF_i\right)$ and $f\left(S_i\right)$ are the fitness of sailfish and sardine.

4 Results and Discussion

4.1 Simulation Setup and Parameters

The dataset from the study of Helwig et al. [25] was used for simulation. In the proposed method, 60% of the data was initially used for teaching, then 20% for validation, and finally the remaining 20% for testing. As discussed before, the Mean Squared Error (MSE) was reduced by setting the goal values identical to the input values and then updating the weights one epoch at a time. There was no predetermined expected outcome during testing and validation. Once the weights were adjusted in training, the output was computed, and the MSE was determined. There were two functions for the validation error. Overfitting was first prevented by constant monitoring. Second, it's employed to determine the cutoff point at which an alarm (a detected attack) was generated if the MSE was found to have been surpassed. Whether the input data were attacked or not was based on the testing error relative to the threshold (for each input). It's important to note that the proposed method required access to neither the fake data nor the labels for the input training data during the training phase. The configured input to the BiGRU included several readings from each sensor at each iteration rather than a sensor, thus using the autocorrelation of domain readings among the sensors. The BiGRU was trained and evaluated for Nt=1, 2, 3, 4, 5, and 10, where Nt is the number of sensors to be supplied to the BiGRU every epoch, in order to determine the optimal number of time instants to be considered. When Nt=2, the training loss was 3.99e-7, the validation loss was 4.37e-7, and the lowest loss value was evident. Both training and validation losses tended to decrease as Nt increased. The data of all 15 sensors were sent to the BiGRU simultaneously. During each iteration, the BiGRU received Nt values from each sensor, which yielded an input layer size of 15Nt neurons. The input and output layers were identical. Five secret layers were used. When an encoder and decoder were used, the compression factor was set to 3, which was the ratio of the number of inputs to the depth of the hidden layer that sat directly between the two. The size of the intermediate layers between the input layer and the deepest layer was shrinked linearly by increasing compression. It's important to note that the compression factor and the number of hidden layers were major determined factors after lots of tests. Table 1 summarizes the BiGRU simulation setting.

4.2 Performance Metrics

To assess the effectiveness of the proposed approach, several measures were considered, including the confusion, accuracy, recall, and accuracy (ACC) matrix and F1 score. The measurements were determined based on the confounding matrix used to measure BiGRU performance.

$$\text{Accuracy } = (TP + TN)/(TP + TN + FP + FN) \tag{16}$$

Table 1. BiGRU simulation parameters

Value	Parameter
20 (Determined by the early stopping standard)	Number of training epochs
Mean of validation error	Threshold
[1 2 3 4 5 10]	Number of readings per sensor (Nt)
Nt * 15	Number of neurons in input layer
Nt * 15	Number of neurons in output layer
3	Density factor
5	Number of hidden layers
Decrease linearly rendering to the density factor	Number of neurons per hidden layer
Unsupervised learning applying backpropagation	Learning algorithm
SOA	Optimizer
0.001	Learning rate
60% of whole set	Training data
20% of whole set	Validation data
20% of whole set + equal number of false data	Testing data

$$\text{Recall} = \text{T P} / (\text{T P} + \text{F N}) \tag{17}$$

$$\text{Precision} = TP/(TP + FP) \tag{18}$$

$$F1 - \text{ measure } = 2 \times (\text{ Precision } \times \text{ Recall }) / (\text{ Precision } + \text{ Recall }) \tag{19}$$

Accuracy and recall were important metrics (i.e., F-score) when combined with unbalanced data. Accuracy referred to the accuracy of the measured outcome and the proximity to the expected solution, while recall was a measure of the number of relevant outcomes. A high recall score represented a low FN, while a high resolution reflected a low FP. Table 2 shows the analysis of various classifiers.

Table 2. Analysis of various classifiers

Method	Accuracy(%)	Recall(%)	Precision(%)	F1-measure(%)
CNN	96.85	11.01	80.15	15.3
RNN	97.34	16.95	84.58	18.72
LSTM	97.9	21.07	90.21	11.14
BiGRU	97.79	35.47	93.22	36.43
BiGRU-SOA	98.5	40.59	96.34	57.95

Table 3. Computational model complexity of various classifiers

Method	Training time (h)	Testing time (m)	Model size (MB)
CNN	28.532	1.32	16
RNN	29.453	1.45	14
LSTM	20.13	1.13	12
BiGRU	21.631	1.25	12.89
BiGRU-SOA	18.145	0.58	11.5

The detection rate increased and the false alarm rate decreased. In addition, the proposed technique had 98% detection percentage. The proposed model had a constant false alarm rate across all scenarios, because the same machine was used for all instances, and was exposed to original (clean) data only during training. However, a freshly trained machine was used for each case, because fraudulent data should be used to train the current models. BiGRU was not taught to categorize any "particular" attack, the proposed technique had the additional benefit of identifying different types of attacks. An attack detector was developed, which identified any attack significantly altering the correlation model of the data. There was no assurance that a CNN, trained to categorize a certain attack, would be

able to identify other attacks. The machine had to be trained with labeled data for every potential attack in order to effectively categorize the attack. Figure 2 and Figure 3 show graphical representation of analysis of the proposed model.

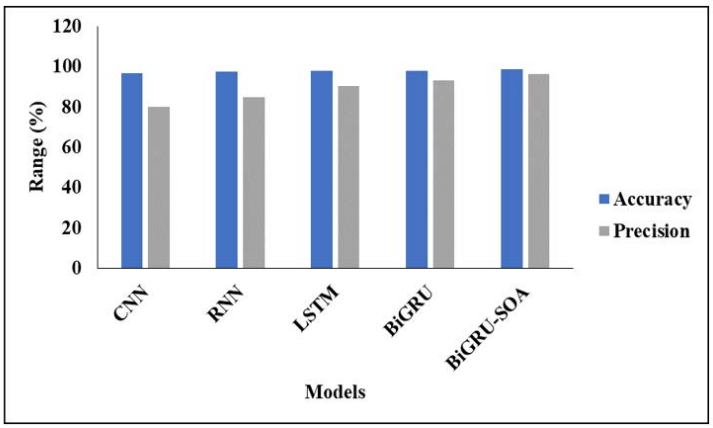

Figure 2. Graphical analysis of the proposed model

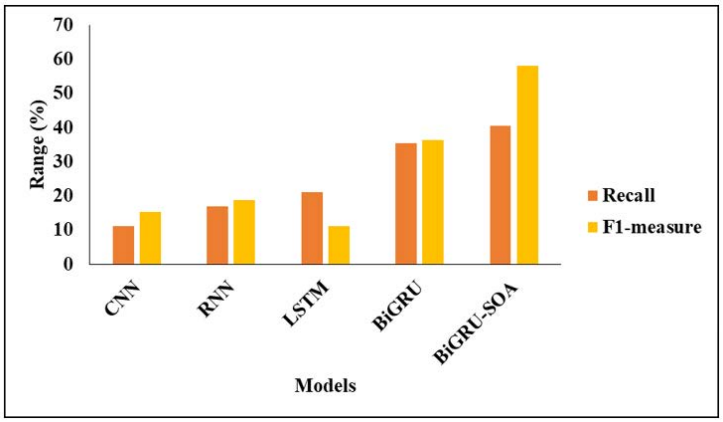

Figure 3. Validation analysis

Table 3 shows the computational complexity of various classifiers. When the models were tested for memory consumption, the consumption was 16MB for CNN, 14MB for RNN, 12MB for LSTM, 12.89MB for BiGRU, and only 11.5MB for the proposed model. This analysis clearly proved that the proposed model achieved better performance than existing models, because its weight was optimally selected by SOA. When training all the input data, the training time was 28hrs for CNN, 29hrs for RNN, 20hrs for LSTM, 21hrs for BiGRU, and only 18hrs for the proposed model.

5 Conclusion

This study proposed a new method to find FDIAs against a complex sensor-based hydraulic system using BiGRU. Furthermore, this method obtained the lowest possible false positive rate (FPR) score since its long-term memory recognized the attack easily. The SOA was applied to the weight selection process of the BiGRU model for maximum efficiency. In addition, training BiGRU was less difficult because labeled data was not needed. Last but not least, BiGRU identified various attacks because it uncovered concealed and complicated correlation patterns in the data. The BiGRU-based attack detection method identified any attack, which significantly altered these correlation structures. Other classifiers, such as CNN, identified only the attack for which they were trained with labeled data, instead of a "particular" attack. In experimental circumstances, the proposed method outperformed the competition in terms of attack detection likelihood, false alarm rate, and runtime. It was also demonstrated how various pre-existing DL models were used to help restore data that was lost in the attack. The findings showed that the proposed BiGRU was quite effective in restoring the data to their original condition with small mean square errors. Hyperparameters can be used to implement the model in the future.

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] H. T. Reda, A. Anwar, and A. Mahmood, "Comprehensive survey and taxonomies of false data injection attacks in smart grids: Attack models, targets, and impacts," *Renew. Sustain. Energy Rev.*, vol. 163, p. 112423, 2022. https://doi.org/10.1016/j.rser.2022.112423

[2] C. Pedroso and A. Santos, "Dissemination control in dynamic data clustering for dense iiot against false data injection attack," *Int. J. Network Management*, vol. 32, no. 5, p. e2201, 2022. http://dx.doi.org/10.1002/nem.2201

[3] R. Vatambeti, N. S. Divya, H. R. Jalla, and M. V. Gopalachari, "Attack detection using a lightweight blockchain based elliptic curve digital signature algorithm in cyber systems," *Int. J. Safety Secur. Eng.*, vol. 12, no. 6, pp. 745–753, 2022. https://doi.org/10.18280/ijsse.120611

[4] Y. Li, X. Wei, Y. Li, Z. Dong, and M. Shahidehpour, "Detection of false data injection attacks in smart grid: A secure federated deep learning approach," *IEEE Trans. Smart Grid*, vol. 13, no. 6, pp. 4862–4872, 2022. https://doi.org/10.1109/TSG.2022.3204796

[5] R. Vatambeti, N. S. Divya, H. R. Jalla, and M. V. Gopalachari, "Attack detection using a lightweight blockchain based elliptic curve digital signature algorithm in cyber systems," *Int. J. Safety Secur. Eng.*, vol. 12, no. 6, pp. 745–753, 2022. https://doi.org/10.18280/ijsse.120611

[6] S. P. Deore, "Human behavior identification based on graphology using artificial neural network," *Acadlore Trans. Mach. Learn.*, vol. 1, no. 2, pp. 101–108, 2022. https://doi.org/10.56578/ataiml010204

[7] H. T. Reda, A. Anwar, A. Mahmood, and N. Chilamkurti, "Data-driven approach for state prediction and detection of false data injection attacks in smart grid," *J. Mod. Power Syst. Clean Energy*, 2022. https://doi.org/10.35833/MPCE.2020.000827

[8] V. K. Damera, R. Vatambeti, M. S. Mekala, A. K. Pani, and C. Manjunath, "Normalized attention neural network with adaptive feature recalibration for detecting the unusual activities using video surveillance camera," *Int. J. Safety Secur. Eng.*, vol. 13, no. 1, pp. 51–58, 2023. https://doi.org/10.18280/ijsse.130106

[9] C. Chen, Y. Wang, M. Cui, J. Zhao, W. Bi, Y. Chen, and X. Zhang, "Data-driven detection of stealthy false data injection attack against power system state estimation," *IEEE Trans. Ind. Inform.*, vol. 18, no. 12, pp. 8467–8476, 2022. https://doi.org/10.1109/TII.2022.3149106

[10] A. P. H., K. K. Almuzaini, L. Ali, A. Javeed, B. Pant, P. K. Pareek, and R. Akwafo, "Delay-driven opportunistic routing with multichannel cooperative neighbor discovery for industry 4.0 wireless networks based on power and load awareness," *Wirel. Commun. Mob. Comput.*, vol. 2022, p. 5256133, 2022. https://doi.org/10.1155/2022/5256133

[11] H. I. Hegazy, A. S. Tag Eldien, M. M. Tantawy, M. M. Fouda, and H. A. TagElDien, "Real-time locational detection of stealthy false data injection attack in smart grid: Using multivariate-based multi-label classification approach," *Energies*, vol. 15, no. 14, p. 5312, 2022. https://doi.org/10.3390/en15145312

[12] H. Tu, Y. Xia, and X. Chen, "Vulnerability analysis of cyber physical systems under the false alarm cyber attacks," *Phys. A Stat. Mech. Appl.*, vol. 599, p. 127416, 2022. https://doi.org/10.1016/j.physa.2022.127416

[13] D. Naga Maheswari, C. B. Sujitha, and K. Ramana, "Routing optimization in sdn using deep reinforcement learning," *J. Eng. Comput. Architecture*, vol. 10, no. 5, pp. 40–47, 2020.

[14] D. Mukherjee, S. Chakraborty, and S. Ghosh, "Deep learning-based multilabel classification for locational detection of false data injection attack in smart grids," *Electr. Eng.*, vol. 104, no. 1, pp. 259–282, 2022. https://doi.org/10.1007/s00202-021-01278-6

[15] M. Q. Tran, M. Amer, A. Dababat, A. Y. Abdelaziz, H. J. Dai, M. K. Liu, and M. Elsisi, "Robust fault recognition and correction scheme for induction motors using an effective iot with deep learning approach," *Measurement*, vol. 207, p. 112398, 2023. https://doi.org/10.1016/j.measurement.2022.112398

[16] S. Acharya, R. Mieth, R. Karri, and Y. Dvorkin, "False data injection attacks on data markets for electric vehicle charging stations," *Adv. Appl. Energy*, vol. 7, p. 100098, 2022. https://doi.org/10.1016/j.adapen.2022.100098

[17] S. Wei, J. Xu, Z. Wu, Q. Hu, and X. Yu, "A false data injection attack detection strategy for unbalanced distribution networks state estimation," *IEEE Trans. Smart Grid*, 2023. https://doi.org/10.1109/TSG.2023.3235945

[18] E. Vincent, M. Korki, M. Seyedmahmoudian, A. Stojcevski, and S. Mekhilef, "Detection of false data injection attacks in cyber–physical systems using graph convolutional network," *Electr. Power Syst. Res.*, vol. 217, p. 109118, 2023. https://doi.org/10.1016/j.epsr.2023.109118

[19] S. Radhoush, T. Vannoy, K. Liyanage, B. M. Whitaker, and H. Nehrir, "Distribution system state estimation and false data injection attack detection with a multi-output deep neural network," *Energies*, vol. 16, no. 5, p. 2288, 2023. https://doi.org/10.3390/en16052288

[20] B. Miao, H. Wang, Y. J. Liu, and L. Liu, "Adaptive security control against false data injection attacks in cyber-physical systems," *IEEE J. Emerging Sel. Topics Circuits Syst.*, 2023. https://doi.org/10.1109/JETCAS.2023.3253483

[21] J. Hua and F. Hao, "Fusion and detection for multi-sensor systems under false data injection attacks," *ISA Trans.*, vol. 132, pp. 222–234, 2023. https://doi.org/10.1016/j.isatra.2022.06.015

[22] Y. Wang, Z. Xu, J. Zhang, L. Xu, H. Wang, and G. Gu, "Srid: state relation based intrusion detection for false data injection attacks in scada," in *19 th European Symposium on Research in Computer Security*. Wroclaw, Poland, Springer, Cham., 2014, pp. 401–418. https://doi.org/10.1007/978-3-319-11212-1_23

[23] X. Liu, Y. Wang, X. Wang, H. Xu, C. Li, and X. Xin, "Bi-directional gated recurrent unit neural network based nonlinear equalizer for coherent optical communication system," *Opt. Exp.*, vol. 29, no. 4, pp. 5923–5933, 2021. https://doi.org/10.1364/oe.416672

[24] S. Shadravan, H. Naji, and V. Bardsiri, "The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 80, pp. 20–34, 2019. https://doi.org/10.1016/j.engappai.2019.01.001

[25] N. Helwig, E. Pignanelli, and A. Schutze, "Condition monitoring of a complex hydraulic system using multivariate statistics," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*. Pisa, Italy, IEEE, 2015, pp. 210–215. https://doi.org/10.1109/I2MTC.2015.7151267