2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)

2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) | 978-1-6654-1502-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/SMARTGRIDCOMM51999.2021.9632305

Defense against Power System Time Delay Attacks via Attention-based Multivariate Deep Learning

Shahram Ghahremani, Rajvir Sidhu, David K.Y. Yau, Ngai-Man Cheung, and Justin Albrethsen
Singapore University of Technology and Design (SUTD)
{ shahram_ghahremani, rajvirkaur_sidhu, david_yau, ngaiman_cheung, justin_albrethsen } @sutd.edu.sg

***Abstract*—Time delay attacks pose a threat to power systems that conventional cybersecurity methods do not adequately address. Conventional methods analyze the contents of network packets to identify threats; this is not effective against time delay attacks, which do not alter packet contents. To detect and identify time delay attacks, a new method is needed. In this paper, a novel and data-driven deep learning (DL) approach is developed to detect time delay attacks on power systems and simultaneously identify both the time of attack and attack magnitude. While conventional DL networks struggle with multivariate long time series data generated by power systems, this can be improved using attention mechanisms. In this paper, dual attention mechanisms (DA) are used to focus and improve a gated recurrent unit (GRU) network for detecting and identifying time delay attacks. A comparative analysis shows the proposed GRU-DA approach outperforms conventional DL, machine learning (ML), and statistical methods while maintaining low model complexity.**

I. Introduction

A cyber-physical system (CPS) utilizes computing and communication technologies for agile monitoring, automation, and control of physical infrastructures. Power systems are a type of CPS whose operation relies on a networked setup of spatially dispersed sensors and actuators [1]. Such connectivity comes with cybersecurity challenges related to protecting communications and data processing.

In a time delay attack, valid and potentially cryptographically protected messages are maliciously delayed in communication links, causing the CPS to use stale information for time-critical control. This can disrupt control loops, leading to system instability and damage to components. Since it is difficult to ensure secure clock synchronization in a distributed CPS, detection and defense against delay attacks are important and urgent research problems. In this paper, we study delay attacks on a power plant control system (PPCS).

A real PPCS has complex configurations with nonlinear correlations among different components, which makes it difficult to create a realistic, tractable, and scalable analytical model. To overcome this challenge, we propose a data-driven approach to detect and identify time delay attacks in a real world PPCS. This approach learns PPCS' properties without prior assumptions, eliminating the need for a complex system model. This method includes a detection mechanism to determine the presence of an attack, and an identification mechanism to elicit the amount of delay and the time of attack.

Recurrent neural networks (RNN) are a type of DL that has shown promising results in time series modeling for CPSes [2]. However, basic RNN is inadequate for long sequence data and has issues with gradients vanishing during training [3]. These problems can be solved with a long-short term memory (LSTM) [4] model and a gated recurrent unit (GRU) [5]. LSTM and GRU have been shown to work in a wide range of tasks such as fault detection [6], solving optimal power flow [7], and electricity load forecasting [8]. A potential concern is that PPCS datasets feature multiple data streams. This allows for mining the underlying system dependencies but may propagate redundant information throughout the learning process. Therefore, there is a need to focus only on relevant information in multivariate time series datasets.

The contributions of the paper are as follows:

- We propose a stacked GRU model to simultaneously predict both the attack point and delay magnitude of time delay attacks against a PPCS. Simultaneous learning is achieved using shared weights in the GRU network.
- We apply dual-stage attention mechanisms integrated with GRU to highlight relevant features while learning on multivariate time series data streams.

II. Related Work

Defending power systems from delay attacks is a relevant problem already considered in recent literature [9]–[11]. Chaudhuri et al. [9] proposed a control design that accounts for normal communication delays in power systems, but this study assumes delays are random and bounded, which may not be true for malicious delays. In [11], a system model is used to estimate delays; unfortunately, model-based analytical methods fail to capture complex relationships between components and may generate complicated or inefficient solutions.

Authors in [12] have proposed an LSTM-based DL technique to characterize time delay attacks; DL performed well if given enough data to learn from. DL was able to construct correlations not shown in training data in scenarios where conventional ML would struggle.

Applying ML and DL to a PPCS is challenging because multivariate and long PPCS data can lead the model to focus

This research was supported in part by the National Research Foundation, Singapore, and the Energy Market Authority, under its Energy Programme (EP award no. NRF2017EWT-EP003-061), and in part by the SUTD-ZJU IDEA Programme (award no. 201805).

on irrelevant parts. To help the model only capture important features, studies have applied attention techniques in their models. Medina et al. [13] proposed parallel attention mechanisms for encoder-decoder networks in a language processing application. The attention mechanism performs well with machine translation but is unproven with PPCS data. In time delay attacks on a PPCS, neural networks struggle to explicitly highlight the relevant input features to predict individual target values. To solve this challenge, this paper proposes a modified dual-stage attention based GRU to detect time delay attacks and identify attack attributes.

III. System Model

In this paper, we consider a Rankine thermodynamic cycle based PPCS, shown in Fig. 1. The PPCS has a generator and two associated control loops, namely a power control loop and an evaporator void control loop [14]. The power control loop regulates the power generated by the generator to a predefined set-point using a PID controller. In a stable system, the PID controller generates the control signal based on the error calculated from the difference between actual power generation and the power set-point. The power set-point is updated as per the load demand. Any delay in control signals generated by the PID controller may lead to an unstable system as power generation may deviate from the set-point.

In this paper, the PPCS is simulated in OpenModelica software and the delay attack is launched against the power control loop. The communication link between the PID controller and the actuators is used to send control signals, which are delayed by the attacker. In general practice, these communication links are not secured and the data is not time stamped. To generate data, simulations are conducted for the PPCS operating in both normal conditions and under delay attacks. The resulting dataset consists of three sensors (power (Pe), pressure (Pr), and temperature (Te)) which describe the power control loop.

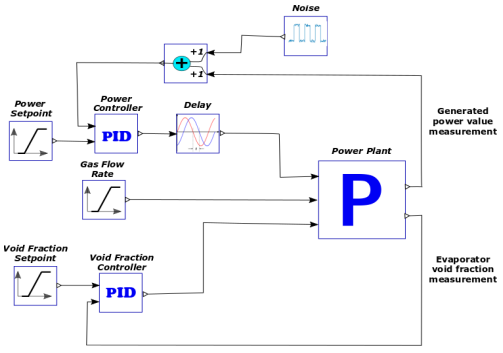


Fig. 1. **A power plant control system simulated in OpenModelica**

A. Time Delay Attack Threat Model

In a PPCS, delay attacks are launched against the power control loop to destabilize the system. The attack may involve jamming communication channels or compromising routers. Attackers can then delay control signals sent from the PID controller to the actuators, by introducing a buffer in the communication line and storing packet values for the desired delay. As the attack is launched, the packets are sent through the buffer in first-in-first-out fashion ensuring that data traffic seems normal, with neither missing nor repeated packets.

Usually, the impact of the delay attack progresses gradually, which is observed in the form of deviations from the normal state of the sensor measurements (Te, Pr, and Pe). If the time delay attack sustains, deviations diverge and lead to an unstable system. Fig. 2 illustrates the trajectory of three sensors' measurements before and after a time delay attack in the power control loop of the PPCS system. Deviations in the sensor measurements immediately after the attack are similar to the inherent (natural) fluctuations and are within stability limits. This similarity makes it challenging for conventional models to detect attacks early. If attacks cannot be identified early it may be difficult to mitigate disturbances in the system.

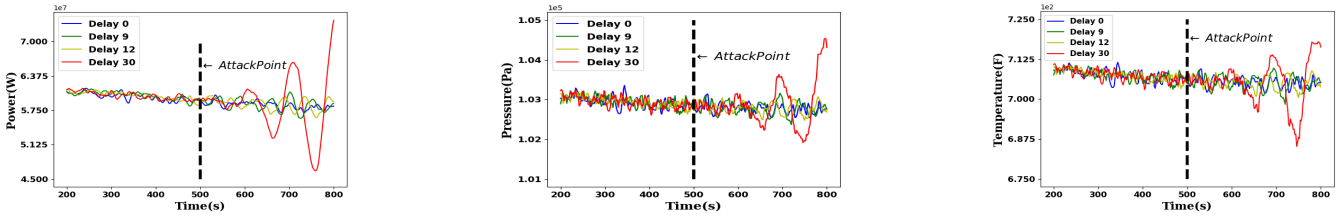
IV. Detection and Identification of Delay Attack

To defend against time delay attacks it is important to know that an attack has occurred, how strong the attack is, and when the attack began. This motivates our primary objectives, to detect the presence of time delay attacks in a PPCS, and to identify key attributes of such an attack. The attributes of interest are the attack starting point, measured in seconds, and the magnitude of the attack, measured in power cycles. For our system, each power cycle lasts for two seconds. To accomplish these detection and identification tasks, we first model a PPCS in OpenModelica as described in section III [15]. Next, we must discover the salient features of the PPCS behavior in the presence of a delay attack. To do that, we run simulations under various time delay attacks and record the system behaviour in the form of sensor measurements. Data from three sensors (Pe, Pr, and Te) are collected during attack simulations; the sensor measurements from all simulations are combined to form a multivariate time series dataset.

For long time series datasets like ours, RNN models such as LSTM or GRU can effectively capture long-term relationships. Of these GRU is preferable as it is simpler than LSTM and may provide quicker predictions, which is very important for real-time systems [16]. However, as explained in section II, these techniques may struggle with multivariate time series data. To address this limitation, we propose using dual attention mechanisms (DA) added to a stacked GRU network. The resultant GRU-DA approach has the benefits of GRU while being able to focus solely on the important input information.

A. Proposed GRU-DA Approach

Our GRU-DA approach attempts to simultaneously solve two problems: attack detection and identification. To detect an attack, the model analyzes a sliding window of sensor measurements and predicts the magnitude of delay within that window. The delay is then compared to a predefined threshold to determine the presence of an attack. The threshold is further described in section V. If the



(a) Power (b) Pressure (c) Temperature

Fig. 2. **The deviations observed in sensor measurements under different time delay attacks**

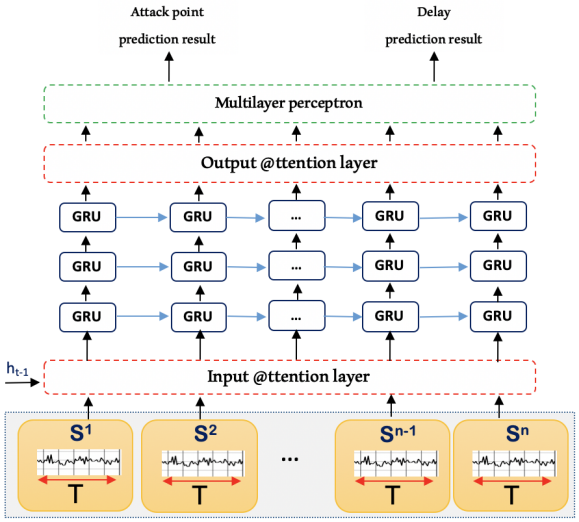


Fig. 3. **The proposed GRU-DA architecture.** S^i **are the input features with window size of T.**

delay prediction is above the threshold then we utilize the second prediction, which identifies at what time the attack started.

The GRU-DA model architecture is illustrated in Fig. 3. The first attention layer extracts the relevant information from the input features at each time step and assigns higher attention weights to the important steps. The weighted input signals are then forwarded to a GRU network with three stacked layers; the output of a GRU hidden layer is fed into the next hidden layer in the subsequent level. Shallower layers capture general features from the input data with shared weights among the layers. Deeper layers extract detailed and more relevant features for each specific target. In the stacked GRU, each layer recombines the learned representation from *prior* layers to create new representations at a higher level of abstraction.

After capturing different kinds of feature patterns, the second output attention layer is applied on top of the stacked GRU layers to assign higher weights to the relevant features across all time steps. As the model tries to predict two targets, attack point and delay magnitude, the second attention layer tries to fine-tune the weights of each time step for each target. The second attention layer also captures the long-term temporal dependencies of a time series appropriately. The output of the last attention layer is then fed into a multilayer perceptron (MLP) layer for regression. The two attention models are well integrated within the stacked GRU layers and can be jointly trained using standard backpropagation. In this way, GRU-DA can adaptively select the most relevant input features as well as capture the long-term temporal dependencies of a multi-variate time series appropriately.

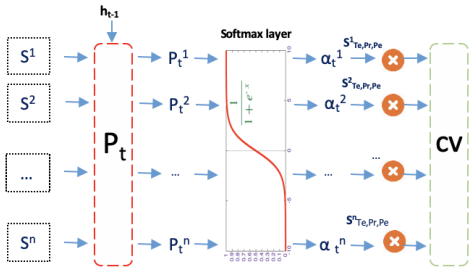


Fig. 4. **Attention Layer Mechanism**

Next we describe the attention mechanisms, which can be visualized in Fig. 4. Given n input features, i.e., $S = (s^1, s^2, \ldots, s^n)^T \in \mathbb{R}^{n \times T}$, where T is the length of window size. We denote $s^k = (s_1^k, s_2^k, \ldots, s_T^k)^T \in \mathbb{R}^T$ to represent the k-th input feature with length T and employ $s_t = (s_t^1, s_t^2, \ldots, s_t^n)_T \in \mathbb{R}^n$ to denote a vector of n input features at time t.

The attention mechanism can be implemented by aggregating a set of vectors s^k into just one vector, often via a lookup vector P_t. We can construct this lookup vector via a multilayer perceptron, by referring to the previous hidden state (h_{t-1}) and the cell state (c_{t-1}) in the encoder LSTM unit as follows:

$$P_t = Q_e^T \tanh(W_e s^k + U_e[h_{t-1}; c_{t-1}]) \quad (1)$$

where Q_e , W_e and U_e are the parameters to be learned.

The probability vector P_t is forwarded to a nonlinear function like Softmax to ensure weights sum to 1 as follows:

$$\alpha_t^k = \frac{\exp(P_t^k)}{\sum_{i=1}^n \exp(P_t^i)} \quad (2)$$

By multiplying the attention weights $\alpha(t)$ and input vectors s_t at the time t, we end up with a vector CV as follows:

$$\text{CV} = (\alpha_t^1 s_t^1, \alpha_t^2 s_t^2, \ldots, \alpha_t^n s_t^n)^T \quad (3)$$

The CV is often called the context vector because it contains the context relevant to all time steps. The CV is then fed into the subsequent GRU layer to assist the prediction task.

B. Simulation Design

The dataset for training and testing the GRU-DA consists of sensor measurements from simulated attacks. Each attack starts and ends between time 220s and 800s and has a random delay magnitude in the range of $[0, 40]$ cycles. We also simulate load changes to represent realistic load demand changes throughout the day. These changes trigger control signal updates sent to the actuator, which can be disrupted by time delay attacks to destabilize the system.

All simulations were run for 800s. We omitted the initial 200s in each simulation because the system takes time to settle down after the startup of the generator. Thus, the sensor data stream shown in Fig. 2 are from time 200s to 800s. Sensor data is logged every 2s, which means we will have 300 data points from a 600s simulation. The complete PPCS sensor dataset consists of 30156 total simulations and is split into training (80%) and test (20%) datasets. To prevent overfitting, we randomly choose 80% of training data to train the GRU-DA and the remaining 20% is for cross-validation.

The performance of the GRU-DA for PPCS delay attack detection and identification is evaluated using three metrics: Root Mean Square error (RMSE), defined as RMSE $= \sqrt{\sum_{i=1}^{N} \frac{(\hat{y}_i - y_i)^2}{N}}$, Mean Absolute Error (MAE), defined as MAE $= \sum_{i=1}^{N} \frac{|\hat{y}_i - y_i|}{N}$, and finally Mean Absolute Percentage Error (MAPE), defined as MAPE $= \frac{100\%}{N} \sum_{i=1}^{N} \frac{|\hat{y}_i - y_i|}{y_i}$ where y_i and $\hat{y}_i$ are the actual and predicted values of the model at the i_{th} test case and N is the total number of test cases. With the RMSE metric, errors are squared before they are averaged, which gives relatively higher weights to large errors. As large errors are particularly undesirable in our case, we use RMSE to optimize our model.

V. Result and Discussion

To detect and identify delay attacks, we leverage our GRU-DA model to predict both the attack point and attack magnitude. Together these predictions satisfy our identification objective. Magnitude predictions can also be used to detect the presence of an attack. We define a threshold ($Th = 2$) for the delay magnitude. If a sample's delay magnitude prediction is less than Th cycles, we treat that sample as having no attack. Observations of time delay attacks on our PPCS show that attacks with delay magnitude less than 2 cycles have negligible impact on the system, so we consider them the same as having no attack. In such cases, the attack point estimates are discarded and will not figure into error measurements.

To show the importance of our GRU-DA, we compare it to both conventional DL and ML methods. These methods include support vector regression (SVR) [17], a machine learning algorithm based on discrete cosine transform (ML-DCT), GRU, LSTM, and GRU with single attention (GRU-IA). It should be noted that all GRU and LSTM models use the same three stacked layers as our GRU-DA, and conventional ML-DCT uses Random Forrest Regression with exponential moving average (EMA) frequencies.

TABLE I
Magnitude prediction results of RNN and ML approaches

Method	MAPE(%)	MAE(s)	RMSE(s)
SVR	84.26	7.79	10.88
ML-DCT	11.92	1.03	2.12
LSTM	6.15	0.53	1.24
GRU	6.06	0.51	1.22
GRU-AI	5.61	0.48	1.11
GRU-DA	**5.10**	**0.40**	**1.10**

TABLE II
Attack point prediction results of RNN and ML approaches

Method	MAPE(%)	MAE(s)	RMSE(s)
SVR	89.77	25.14	41.34
ML-DCT	14.14	5.72	8.41
LSTM	9.14	3.78	6.32
GRU	9.01	3.61	6.21
GRU-AI	7.89	3.11	5.65
GRU-DA	**7.11**	**2.88**	**4.90**

First, we examine our performance in identifying the key parameters of the attacks. The performance metrics for detecting delay magnitude are included in Table I and the attack point metrics are shown in Table II.

Similar trends are found in both Table I and Table II and will be discussed jointly. Large values of RMSE, MAE, and MAPE for ML-DCT and SVR suggest that conventional ML is unable to abstract features as well as the DL techniques. ML underperforms because it cannot remember historical information from time series data like our PPCS dataset. In contrast, DL models like LSTM and GRU can remember relevant historical data and capture temporal context. Among DL techniques, the ones with attention layers outperform conventional stacked DL techniques (GRU and LSTM). This suggests that the attention mechanisms do prioritize important driving series and give more reliable input features to the GRU. Furthermore, the proposed GRU-DA, with an additional attention layer applied for the output, exceeds all baseline methods

TABLE III
Attack detection results of RNN and ML approaches

Method	AUC(%)	FPR(%)	TPR(%)
SVR	76.5	19.5	80.7
ML-DCT	81.3	7.39	83.9
LSTM	96.3	0.43	97.4
GRU	96.4	0.43	97.6
GRU-AI	97.1	0.29	98.0
GRU-DA	**97.8**	**0.21**	**98.6**

for both tasks.

Next, we examine how the same techniques perform with attack detection. For these experiments, we consider a sample to be a false positive if the delay magnitude prediction is above our detection threshold, but the ground truth delay is below it. Similarly a true positive represents a sample containing a true delay above the threshold, and the predicted delay is also above the threshold. The false positive rate (FPR), true positive rate (TPR), and the area under the curve (AUC) are shown in Table III. The curve for the AUC measure is generated by plotting the TPR against the FPR, also known as the receiver operating characteristic (ROC) curve.

Since attack detection hinges on identification of attack magnitudes, we expect similar trends for both sets of results. This is true, as the performance rankings for attack detection mirror the rankings for our delay magnitude predictions. We can see that for all metrics, our proposed GRU-DA continues to outperform all other techniques. The results show that GRU is the highest performing generic technique, and each attention mechanism further enhances performance for both detecting and identifying time delay attacks.

A. Analysis of Attention Mechanisms

The attention mechanisms described in section IV-A are implemented within a three layer GRU network. These mechanisms highlight the important input series within time steps by distributing attention weights. The distribution of attention weights is visualized with heat maps in Fig. 5. The heat maps in this figure represent four scenarios, for each scenario, the attack point is fixed and the delay magnitude is varied. The attack point is fixed in time step 19 (5a), time step 16 (5b), time step 9 (5c), and time step 3 (5d). Higher attention weights are given to the time steps shortly after the attack, indicating that the attention mechanisms prioritize immediate post-attack data. For each scenario, weight distribution among the time steps shifts with the attack point. The only exception is the scenario with no post attack data (5a) in which the attention weights are nearly homogeneous. This shows that attention mechanisms focus on the post-attack time steps and illustrates the importance of having post attack data.

B. GRU-DA Hyperparameter Optimization

The GRU-DA has hyperparameters which must be manually set and may affect the model performance. We study the sensitivity of GRU-DA with respect to its hyperparameters, specifically the number of time steps *TS*, the number of stacked layers *L*, and input window length.

Grid search is performed to find the optimal number of time steps. Since our magnitude prediction is used for attack detection, this is considered a higher priority and will be used for our optimization. Using a constant three stacked layers, we plot the RMSE versus different values of TS in Fig. 6 (a). The RMSE is calculated for the delay magnitude prediction only. The results show that the GRU-DA and GRU achieve the lowest RMSE when TS=20.

TABLE IV
ATTACK DETECTION RESULTS OF RNN APPROACHES ON VARIOUS WINDOW LENGTHS.

Method	AUC(%)			FPR(%)		
	200	***300***	***400***	***200***	***300***	***400***
LSTM	92.34	96.3	79.57	0.52	0.43	0.60
GRU	92.58	96.4	79.12	0.51	0.43	0.62
GRU-DA	94.01	97.8	82.21	0.41	0.21	0.55

To see the impact of the number of stacked layers on GRU-DA, we vary the number of layers while keeping other hyperparameters constant. Keeping TS=20, we plot the RMSE evaluation metric against the number of layers, as seen in Fig. 6 (b). All three approaches achieve the best performance when L=3. Adding more layers to the model does not necessarily improve the performance. This is because aggressively increasing the capacity of the model results in over-fitting to the training data.

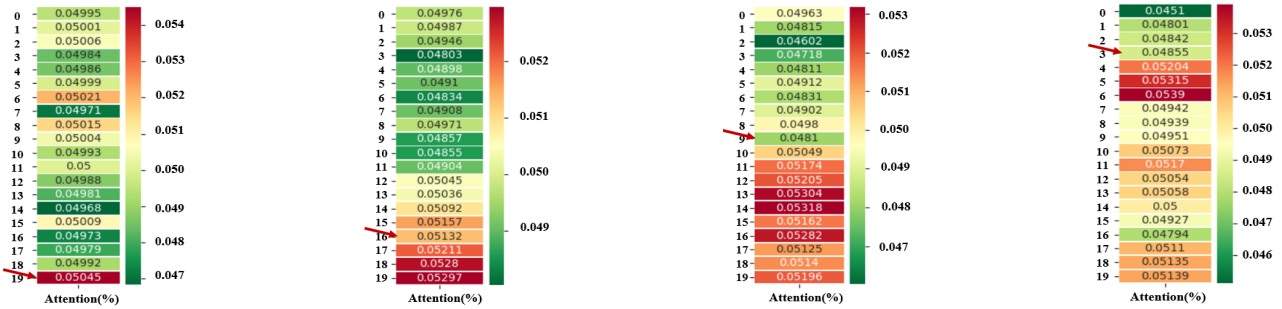
Finally we select the length of our input windows. We try 3 different window lengths of 200, 300, and 400 data points. In Table IV we compare the performance of our model and other conventional DL techniques with varying window length. For all models, the performance with 300 data-points is better than other window lengths. We infer that as we use a long window, the number of multivariate data points which belongs to the same time window increases, which can lead to the vanishing gradients problem in a long-term sequence. On the other hand, too short window length cannot capture the patterns of the time windows with statistically aggregated values. We see that input window length is an important hyper parameter that impacts our performance.

TABLE V
COMPARISON OF MODEL COMPLEXITY FOR LSTM, GRU, AND GRU-DA

Method	# of Parameters	Testing Time (s)
LSTM	5347001	0.6133
GRU	4073001	0.4975
GRU-DA	**4074861**	**0.5286**

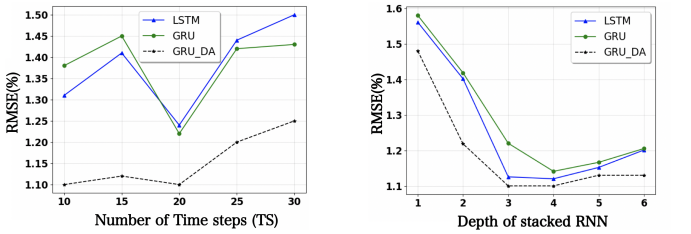
C. Model Complexity

Finally, the comparison of model complexity for three layers of LSTM, GRU, and GRU-DA is illustrated in Table V. The number of parameters is a measure of complexity taken from the Keras model summary [18] and represents the total weights and biases in the fully trained model. The testing time measures the time taken to make a single prediction. The table shows that GRU uses fewer parameters than LSTM, which corresponds with a shorter training time. Adding the attention layers to the GRU incurs few additional parameters to the model and the additional overhead may be considered negligible. It can also be observed that testing time for GRU is comparatively shorter than LSTM, which is important for applications that demand recurring and timely predictions.



(a) Attack starts in step 19 (b) Attack starts in step 16 (c) Attack starts in step 9 (d) Attack starts in step 3

Fig. 5. **Heat map for weight distribution for different time steps of GRU-DA. Red arrows indicate the attack starting point.**



(a) RMSE vs. number of time steps TS (b) RMSE vs. number of RNN stacked layers L

Fig. 6. **The impact of (a) number of time steps, and (b) number of RNN stacked layers on estimating delay attack magnitude.**

VI. Conclusion

This paper presents a novel data-driven DL approach to detect and identify time delay attack attributes in the control communication line of a PPCS. The dual attention mechanism is developed and embedded in the stacked GRU layers to extract underlying dependencies among the time series data and predict both the delay magnitude and the time of attack. Input and output attention layers adaptively adjust the weights of the input and features extracted at the output of GRU, respectively, to capture the most relevant features and ignore irrelevant information within the data series. Comparative analysis shows that GRU-DA outperforms other ML and DL techniques for both attack detection and attack attribute identification, while maintaining low model complexity.

References

[1] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[2] Y. Fu, F. Lou, F. Meng, Z. Tian, H. Zhang, and F. Jiang, "An intelligent network attack detection method based on rnn," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2018, pp. 483–489.

[3] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[4] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000, publisher: MIT Press.

[5] N. Wang, J. Wang, and X. Zhang, "YNU-HPCC at SemEval-2018 Task 2: Multi-ensemble Bi-GRU Model with Attention Mechanism for Multilingual Emoji Prediction," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 459–465.

[6] P. Tehrani and M. Levorato, "Frequency-based Multi Task learning With Attention Mechanism for Fault Detection In Power Systems," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2020, pp. 1–6.

[7] D. Urgun and C. Singh, "LSTM Networks to Evaluate Composite Power System Reliability Evaluation with Injected Wind Power," in *2019 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2019, pp. 1–5.

[8] S. Liu, C. Xu, Y. Liu, D. Katramatos, and S. Yoo, "Electricity Load Forecasting with Collective Echo State Networks," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2020, pp. 1–6.

[9] B. Chaudhuri, R. Majumder, and B. C. Pal, "Wide-area measurement-based stabilizing control of power system considering signal transmission delay," *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1971–1979, 2004, publisher: IEEE.

[10] X. Lou, C. Tran, R. Tan, D. K. Yau, and Z. T. Kalbarczyk, "Assessing and mitigating impact of time delay attack: a case study for power grid frequency control," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, 2019, pp. 207–216.

[11] A. Sargolzaei, K. K. Yen, and M. N. Abdelghani, "Preventing time-delay switch attack on load frequency control in distributed power systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1176–1185, 2015.

[12] X. Lou, C. Tran, D. K. Yau, R. Tan, H. Ng, T. Z. Fu, and M. Winslett, "Learning-Based Time Delay Attack Characterization for Cyber-Physical Systems," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2019, pp. 1–6.

[13] J. R. Medina and J. Kalita, "Parallel attention mechanisms in neural machine translation," in *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 547–552.

[14] "ThermoPower." [Online]. Available: https://casella.github.io/ThermoPower/

[15] "OpenModelica." [Online]. Available: https://www.openmodelica.org/

[16] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," in *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*. IEEE, 2020, pp. 98–101.

[17] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155–161.

[18] K. Team, "Keras documentation: The Model class." [Online]. Available: https://keras.io/api/models/model/summary-method