Journal of Network Intelligence
Taiwan Ubiquitous Information

Volume **9**, Number **1**, February **2024**

An Intelligent Perception Model for DC Transmission Line Live Working Risk Prediction based on Deep Learning

Jingxiang Li*, Zhibo Jiang, Hao Lai, Haitao Yin

EHV Power Transmission Company, CSG, Guangzhou 510006, China
ljx.hust@163.com, jiangzhibo@im.ehv.csg, 254901390@qq.com, 835433916@qq.com

*Corresponding author: Jingxiang Li
Received September 3, 2023, revised November 17, 2023, accepted January 23, 2024.

Abstract.
In the realm of electrical safety, live working on high-voltage direct current (DC) transmission lines poses formidable challenges and risks. This paper introduces the "LW-Risk Prediction" (LW stands for live working), an innovative intelligent perception model based on deep learning designed to improve the prediction of risks in the context of DC transmission line live working. There are two main challenges in DC transmission line live working risk prediction. One is that the DC transmission line live working risk has many relative factors such as weather, transmission line fault, and attack events etc. And the state of these factors is changing over time. The other one is that the risk of transmission line live working may be fatal and the prediction should be timely. This paper explores the model's innovative design elements, including domain-specific expertise, temporal modelling using recurrent neural networks (RNNs) with attention mechanisms, and an embedding approach to leverage both categorical and numerical data. According to the experiment results, the proposed model can utilize many relative factors and give a high accuracy and timely prediction for DC transmission line live working risk.
Keywords: Risk Prediction, Live Working, Deep Learning, Transmission Line

---

1. **Introduction.** Live working for direct current (DC) transmission lines means efficient and continual maintenance without disconnecting the electricity supply from the consumers [1]. Compared with the maintenance requiring disconnection of parts of the electricity network, live working can save millions of pounds and therefore become widespread in electricity transmission line maintenance. However, when working with a live line, linesmen are exposed to high electromagnetic fields and take a high risk which can be fatal. To ensure the reliability and safety of live working on high voltage DC transmission lines, many researchers focus on the protection of the live working linesmen [2–4]. Moreover, live line working standards were also present such as IEC 61472 [5], IEEE 516-1995 [6].

Even though, the risk of live working on DC transmission lines is still at a high level and can not be well-predicted. One of the reasons is that there are various factors influencing the safety of live working, such as the air gap insulation, the weather, the faults of the transmission lines etc [7]. And those factors may change over time. The safety of the live working linesmen can be adversely affected by these under certain conditions. And live line working standards in some degrees give the safety criteria by assuming the factors are statical. For example, in a given condition (weather, voltage, distance) at some moment, a lineman conducting live work is supposed to be safe. However, when the condition changes, the risk of this live working should be warned as soon as possible. Thus, it is essential to predict the risk of the live working for direct current (DC) transmission lines timely.

Although risk prediction is vital in live working for direct current (DC) transmission lines, few methods take diverse factors into account when studying this problem. Many of them only focus on a single key parameter like the approach distance. For example, [8] provides the prediction of breakdown voltage which affects the calculation of the distance of the equipotential live-line work air gap. [9] concerns the potential transfer of current and arc energy. Integrating the relative factors as much as possible becomes the main challenge of the risk prediction on live working for direct current (DC) transmission lines. Traditionally, methods based on big data frequent item mining [10, 11] can deal with the monitoring data with high complexity and make an early warning. However, since many of the relative factors are changing over time, it is still difficult for a traditional method to predict the risk timely. Due to the advantage of capturing complex and non-linear relationships between predictors and outcomes by learning from large-scale and heterogeneous data sources, deep learning has become popular and applied in many scenarios these years. In this paper, we provide an innovative way based on deep learning and try to address the aforementioned challenges on the risk prediction on live working for direct current (DC) transmission lines. Specifically, our contributions are summarized as follows:

- We propose an innovative deep learning model to predict the risk of live working on high-voltage direct current (DC) transmission lines. Compared with the traditional data mining method, the proposed model is able to take diverse relative factors into account and thus make a more reasonable prediction.
- Since most of the relative factors about live working on high-voltage DC transmission lines will change over time, it is difficult for traditional methods like big data frequent item mining to give timely early risk warnings. Contrastly, the proposed model can capture the inner relationship of those relative factors after training and make a timely prediction.
- In our experiments, we not only conduct the test on the proposed model but also on several classical deep learning models over the same dataset. From the given experimental results, the proposed model shows a better performance than these well-known deep learning models.

The remaining sections are arranged as follows. Section 2 introduces the related research on live working risk prediction. Section 3 explains the data preprocessing, the training and validation process for the model. Section 4 describes the architecture of the proposed model in detail. Section 5 presents the results of the model's performance, including accuracy and relevant metrics. Finally, Section 6 makes a summary of the work.

2. **Related work.** Risk prediction for linesmen who work on live direct current (DC) transmission lines is a challenging and important problem, as it involves both technical and human factors. Risk assessment methods aim to identify and quantify the potential hazards and consequences of working on live DC lines. There are some methods in common use, including fault tree analysis [12], event tree analysis [13], Bayesian networks [14], and fuzzy logic [15]. These methods can help to evaluate the probability and severity of different scenarios, such as electric shock, arc flash, fire, or equipment failure. However, these methods also have some limitations, such as data availability, uncertainty, and complexity. In 2018, Qiu et al. proposed an approach for discharge voltage prediction of complex gaps based on support vector machine (SVM) [16]. In 2021, [17] constructed

digital twin bodies of live workers in different live working scenarios for risk assessment. In 2022, Sun et al. [11] provided a risk early warning method based on big data frequent itemset mining which can deal with power monitoring data with high complexity. However, integrating data with high diversity and giving a timely prediction at the same time is still a challenge and vital for live working on DC transmission lines.

3. **Methodology.** The methodology section details the steps taken in the development, training, and evaluation of the "LW-Risk Prediction" model for predicting risks in the context of DC transmission line live working.

3.1. **Data Synthesis.** The dataset we used in this paper can be accessed in [18]. The "LW-Risk Prediction" model development chooses a strategy of synthesized data which mimics the distribution of real-world data. This comprehensive approach aims to create a robust and effective model while addressing the challenges of data availability and diversity. Synthetic data plays a pivotal role in the initial stages of model development. It provides a controlled environment for training and experimentation. The synthetic dataset is created with meticulous attention to mimic real-world characteristics while ensuring controllability. The data Synthesis algorithm is outlined below:

Algorithm 1 Data Synthesis

```
start_date = Datetime(2020, 1, 1, 0, 0)
delta = Timedelta(minutes=5)
num = 315360
voltages = RandomChoice([400, 500, 660, 800, 1100], size=num)
temperature_mean, temperature_std = MimicFromRealWorld()
humidity_mean, humidity_std = MimicFromRealWorld()
weather_conditions = MimicFromRealWorld()
distance_mean, distance_std = 7.5, 1.5
temperatures = RandomChoice(temperature_mean, temperature_std, num)
humidities = RandomChoice(humidity_mean, humidity_std, num)
distances = RandomChoice(distance_mean, distance_std, num)
winds = RandomChoice([0, 1, 2, ..., 17], size=num_samples)
types_of_work= RandomChoice(['Equal potential', 'Isolation', 'Zero potential'],
    size=num)
abnormality = RandomChoice([0, 1, 2], size=num, p=[0.8, 0.1, 0.1])
for i in Range(num - 1) do
    irow = GetFeatures(i)
    iscore = RiskScoreVariation(irow)
    threshold = 25
    risk[i] = (iscore > threshold)
end for
```

As is shown in algorithm 1, the synthesis process involves:

- **Temporal Dynamics**: Capturing time-dependent patterns and dynamics, such as time of day and date, to simulate the temporal aspect of the data. We choose a range from 2020-1-1 to 2022-12-31 and mimic the distribution feature from the real-world weather condition data.
- **Feature Generation**: Generating features that resemble those found in real-world data, including live-line work air distance, weather conditions, voltage levels, and work-related attributes.

Table 1. Sample Data for "LW-Risk Prediction" Model

Date	**2020/1/1**	**2021/1/1**	**2020/2/1**	**2020/3/1**	**2020/4/1**
Time	00:00	00:05	00:10	00:15	00:20
Voltage	400 kV	500 kV	660 kV	800 kV	1100 kV
Weather	Rainy	Cloudy	Foggy	Thunder	Sunny
Humidity	52	88	49	67	69
Wind(Level)	0	1	2	3	4
Distance (m)	7	8	3	5	6
Abnormality	0	1	2	1	2
Temperature (°C)	21	23	30	28	26
Work Type	Zero potential	Equal potential	Isolation	Isolation	Isolation
Risk	0	1	1	1	1

- **Anomaly Scenarios**: Introducing anomalous events and scenarios to test the model's ability to detect and respond to unusual circumstances.
- **Data Variation**: Creating variations in data distributions and risk relationships to ensure the model's adaptability to diverse conditions.

3.2. **Data Preprocessing.** Data preprocessing is a critical step to ensure the model's input data is suitable for training. The sample data is given in Table 1. The data preprocessing algorithm is outlined below:

Algorithm 2 Data Preprocessing Algorithm

```
reader = open('dataset')
newcolumns = []
DateTime = [reader.column[0],reader.column[1]]
newcolumns.append(DateTime)
for i in range(2, reader.columNumber) do
  if reader.column[i] is categorical variables then
    reader.column[i] = one-hot(reader.column[i])
  else
    reader.column[i] = Standardize(reader.column[i])
  end if
  newcolumns.append(reader.column[i])
end for
return newcolumns
```

As is shown in algorithm 2, the data preprocessing steps were performed as follows:

- **Temporal Features**: The "Date" and "Time" columns were utilized to extract relevant temporal features, such as the time of day and the day of the week, enhancing the model's understanding of time-dependent patterns.
- **Categorical Variables**: Categorical variables, including "Voltage", "Weather", "Wind Level", "WorkType" and "Abnormality" were transformed using one-hot encoding, converting them into a numerical format.
- **Numerical Variables**: Numerical variables, specifically "Humidity", "Temperature", and "Distance" were standardized to have a mean and a standard deviation which mimic the real-world data. Standardization helps the model converge faster during training and ensures consistent scaling.

3.3. **Training and Validation Process.** The training and validation process for the "LW-Risk Prediction" model is a critical phase in its development. It involves several key steps to ensure the model's robustness and reliability:

- **Data Split**: The dataset is divided into two subsets: the training set, and the test set. The typical split ratio used is 80% for training, 20% for testing. This partitioning ensures that the model is trained on a substantial portion of the data while having separate datasets for model evaluation.
- **Model Training**: The training phase involves optimizing the model's weights and biases to minimize a suitable loss function. The choice of loss function depends on the specific objectives of the model. The proposed model uses binary cross entropy as its loss function. Binary cross entropy is a loss function used in binary classification problems, where the goal is to predict the probability of an outcome that can be either 0 or 1. It measures how close the predicted probabilities are to the true labels, and penalizes incorrect predictions with a higher loss value. The formula for binary cross entropy is

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

  During training, backpropagation is used to update the model's parameters iteratively. The training process is guided by a well-defined stopping criterion to prevent overfitting. It may halt when the model achieves satisfactory performance after a predetermined number of training epochs.
- **Hyperparameter Tuning**: Hyperparameters play a crucial role in model performance. Parameters like the learning rate, batch size, and network architecture are fine-tuned to optimize the model's predictive accuracy. Hyperparameter tuning is typically conducted using techniques like grid search or random search to explore different combinations of hyperparameters. There are six key hyperparameters to be tuned in the proposed model. They are the number of epochs, the number of batch sizes, the learning rate, the data training ratio, the type of RNN (LSTM or GRU) and the number of units in the chosen RNN. To find a group of suitable hyperparameters for the proposed model, we give a range for each hyperparameter like a grid search. Then we tune one of the hyperparameters each time by fixing the value of the other hyperparameters and observing the performance of the model when changing the value of the tuning hyperparameter.

4. Model Architecture.

4.1. **Overview.** The "LW-Risk Prediction" model is a sophisticated deep learning architecture that excels in predicting risks associated with live working on high-voltage DC transmission lines. It integrates both categorical and numerical data to provide a comprehensive risk assessment. The overview architecture is given in Figure 1. Below, we describe the principles and equations for each layer of the model.

4.2. **Embedding Layers.** An embedding layer is a type of hidden layer in a neural network that maps input information from a high-dimensional to a lower-dimensional space, allowing the network to learn more about the relationship between inputs and to process the data more efficiently. Categorical features in the given dataset such as "Wind Level", "Voltage", "Weather", "WorkType" and "Abnormality" are initially processed through embedding layers to convert them into continuous representations. To use an embedding layer, we need to first integer encode the input data, so that each word or

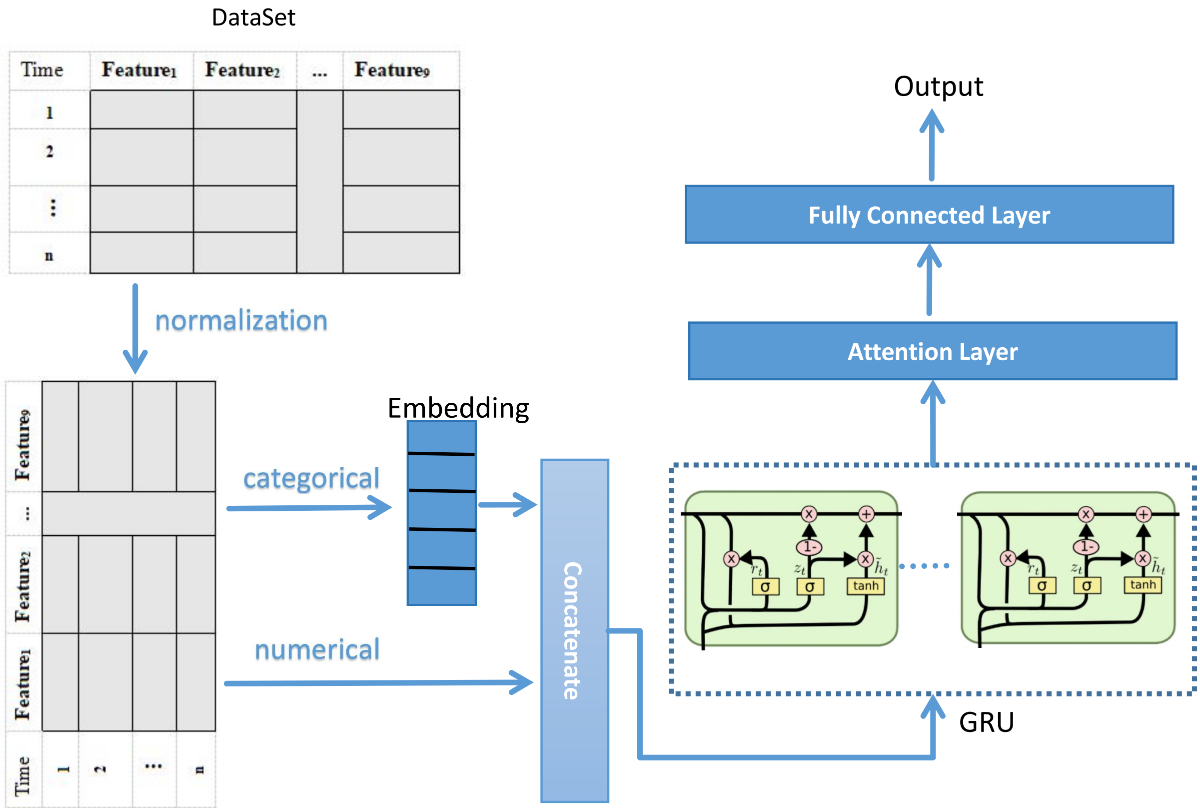

FIGURE 1 The Proposed Model

category is represented by a unique integer. For example, we can assign the word "Sunny" to the integer 1, and the word "Rainy" to the integer 2. Then, we can create an embedding layer with a specified input dimension (the size of the vocabulary), output dimension (the size of the embedding vector), and input length (the length of the input sequence). The embedding layer will initialize a matrix of weights with random values, where each row corresponds to an embedding vector for a word or category. During training, the embedding layer will update the weights to learn the optimal representation for each word or category. The embedding layer for a categorical feature "X" is defined as follows:

$$\text{Embedding}_X = \text{Embedding}(X, \text{input\_dim}_X, \text{output\_dim}_X) \tag{1}$$

Where: Embedding_X represents the embedding layer for feature "X". $\text{Embedding}(X, \text{input\_dim}_X, \text{output\_dim}_X)$ denotes the embedding function for feature "X" with input dimension input_dim_X and output dimension output_dim_X.

4.3. **Concatenate Layer.** A concatenate layer is a type of layer in a neural network that combines a list of inputs into a single output. It takes as input a list of tensors, all of the same shape except for the concatenation axis, and returns a single tensor that is the concatenation of all inputs along that axis. A concatenate layer can be useful for combining different types of features or information from different sources. In the proposed model, we use a concatenate layer to combine the categorical features after the embedding layers with the numerical features.

4.4. **Gated recurrent units(GRU) Layer.** A Gated recurrent units (GRU) [19] model is a type of recurrent neural networks (RNNs) [20] that uses gating mechanisms to selectively update the hidden state at each time step, allowing it to effectively model sequential data. A GRU model has two gates: a reset gate and an update gate. The reset gate determines how much of the previous hidden state should be forgotten, while the update gate determines how much of the new input should be used to update the hidden state. The output of the GRU model is calculated based on the updated hidden state. The core units for a GRU model are as follows:

$$\text{Reset gate:} r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{2}$$

$$\text{Update gate:} z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{3}$$

$$\text{Candidate hidden state:} \tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t]) \tag{4}$$

$$\text{Hidden state:} h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{5}$$

where σ is the sigmoid function, $\odot$ is the element-wise product, W_r, W_z, and W_h are learnable weight matrices, x_t is the input at time step t, h_{t-1} is the previous hidden state, and h_t is the current hidden state.

4.5. **Attention Layer.** The attention layer is a way of enhancing the performance of a recurrent neural network (RNN) by allowing it to focus on the most relevant parts of the input sequence. The attention layer takes the output of a GRU layer and computes a weighted sum of the hidden states of the GRU. The weights are learned by a dense layer with a tanh activation function, which assigns a score to each hidden state. The scores are then normalized by a softmax function, which produces a probability distribution over the hidden states. The attention layer then outputs a vector that is the dot product of the probability distribution and the hidden states, which represent the most important features of the input sequence. The output of the attention layer can then be fed to another dense layer with a sigmoid activation function, which produces the final prediction of the model. The following formulas show the mathematical operations of the attention layer:

$$\text{GRU output: } \mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_T \in \mathbb{R}^d \tag{6}$$

$$\text{Attention scores: } \mathbf{s} = \tanh(\mathbf{W}_a \mathbf{H} + \mathbf{b}_a) \tag{7}$$

$$\text{Attention weights: } \mathbf{a} = \text{softmax}(\mathbf{s}) \tag{8}$$

$$\text{Attention output: } \mathbf{c} = \mathbf{a}^\top \mathbf{H} \tag{9}$$

where $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_T]^\top \in \mathbb{R}^{T \times d}$ is the matrix of GRU hidden states, $\mathbf{W}_a \in \mathbb{R}^{1 \times d}$ and $\mathbf{b}_a \in \mathbb{R}$ are the parameters of the dense layer, $\mathbf{s} \in \mathbb{R}^T$ is the vector of attention scores, $\mathbf{a} \in \mathbb{R}^T$ is the vector of attention weights, and $\mathbf{c} \in \mathbb{R}^d$ is the vector of attention output.

4.6. **Fully Connected Layer.** The fully connected layer processes the flattened data, applying a linear transformation followed by a ReLU activation function:

$$Z = \text{ReLU}(W_f X + b_f) \tag{10}$$

Where: Z represents the output of the fully connected layer. W_f is the weight matrix. X is the input data. b_f is the bias term.

4.7. **Output Layer.** The output layer is responsible for producing the final risk prediction. It uses a sigmoid activation function to produce a probability score between 0 and 1:

$$\text{Risk Prediction} = \sigma(W_o Z + b_o) \tag{11}$$

Where: Risk Prediction represents the final risk prediction. W_o is the weight matrix for the output layer. Z is the input from the fully connected layer. b_o is the bias term.

The sigmoid activation function σ is used to squash the output into the range [0, 1], providing the probability of risk associated with live working on DC transmission lines. A threshold can be set to categorize the risk as low, medium, or high based on the predicted probability.

5. **Results and Discussion.** There are two metrics in our experiment to evaluate the performance of the proposed model. One is the prediction accuracy. The other one is Area Under the ROC Curve (AUC) [21].

AUC is a metric that summarizes the performance of a binary classifier across all possible thresholds. It measures how well the classifier can distinguish between the positive and negative classes, regardless of the actual class distribution or the chosen threshold. The AUC is equal to the probability that the classifier will assign a higher score to a randomly selected positive example than to a randomly selected negative example. The AUC ranges from 0 to 1, where 0 means the classifier is completely wrong and 1 means the classifier is completely right. A random classifier would have an AUC of 0.5, which means it cannot discriminate between the classes at all. One way to calculate the AUC is to use the trapezoidal rule to approximate the area under the ROC curve, which is a plot of the true positive rate (TPR) versus the false positive rate (FPR) at different thresholds. The TPR and FPR are defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

where TP, FP, FN, and TN are the number of true positives, false positives, false negatives, and true negatives, respectively. To apply the trapezoidal rule, we first sort the classifier scores in descending order and compute the corresponding TPR and FPR values for each score as a potential threshold. Then, we divide the ROC curve into a series of trapezoids, each with a base equal to the difference between two adjacent FPR values and a height equal to the average of the corresponding TPR values. The area of each trapezoid is given by:

$$A_i = \frac{(FPR_{i+1} - FPR_i) \times (TPR_{i+1} + TPR_i)}{2}$$

where i is the index of the score. The AUC is then the sum of the areas of all the trapezoids:

$$AUC = \sum_{i=1}^{n-1} A_i$$

where n is the number of scores.

Generally, we tune the above hyperparameters by changing the value of one parameter and fixing the values of the other parameters to see the different results both on prediction accuracy and AUC. As is shown in Figure 2 to 6, we give the experiment results in different hyperparameters, namely the training ratio, the learning rate, the number of units in GRU, the number of batch size and the number of epoch.

5.1. **Experiment Tuning.** As has been mentioned, there are six key hyperparameters to be tuned in the proposed model. Firstly, we choose GRU instead of LSTM as a component of the proposed model according to the experiment results. The remaining hyperparameters are tuned as follows:

The number of epochs is the number of times that the entire training data is passed through the model. A higher number of epochs means more exposure and learning for the model, but also more risk of overfitting and longer training time. As is shown in Figure 2, the performance becomes better as the number of epochs increases.

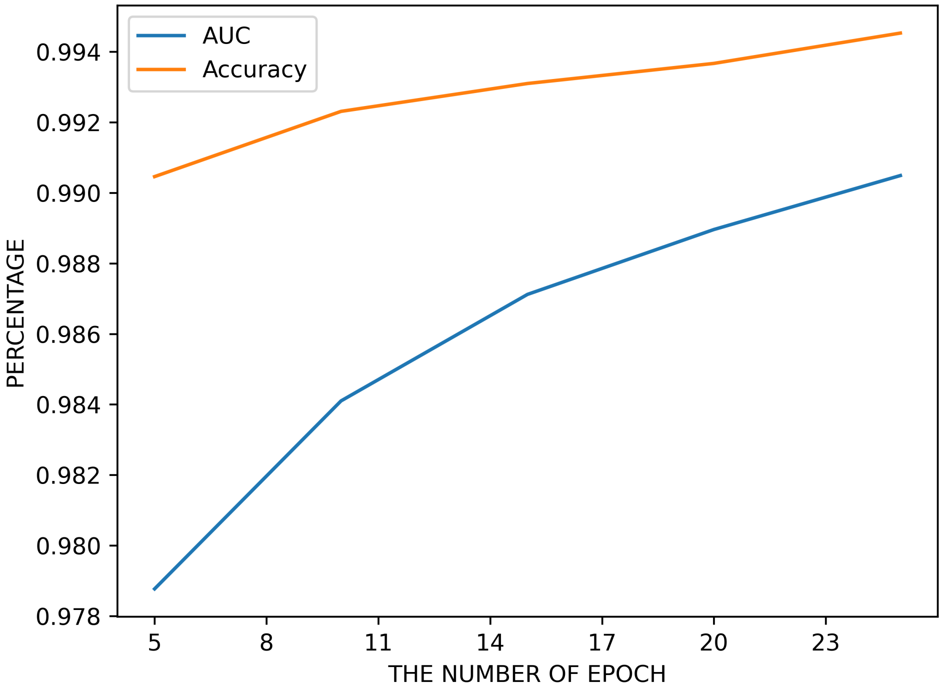

FIGURE 2 Different Number of Epoch

The batch size is the number of samples that are processed together in each iteration of the training. A higher batch size means more parallelism and faster convergence, but also more memory usage and less stochasticity. As is shown in Figure 3, a recommended batch size is about 96.

The learning rate controls how much the model's weights are adjusted with respect to the loss gradient. If the training rate is too high, the model might overshoot the optimal point. If the training rate is too low, the model might get stuck in a local minimum or take too long to converge. As is shown in Figure 4, a recommended learning rate is about 0.005.

The parameter Units in a GRU model specifies the number of hidden units in each GRU cell. The higher the number of units, the more expressive and complex the GRU model can be, but also the more parameters and computational cost it will have. As is shown in Figure 5, a recommended range is in $(16, 20)$.

The training ratio is the proportion of the data that is used for training the model, while the rest is used for testing. As is shown in Figure 6, an 80/20 split is suggested,

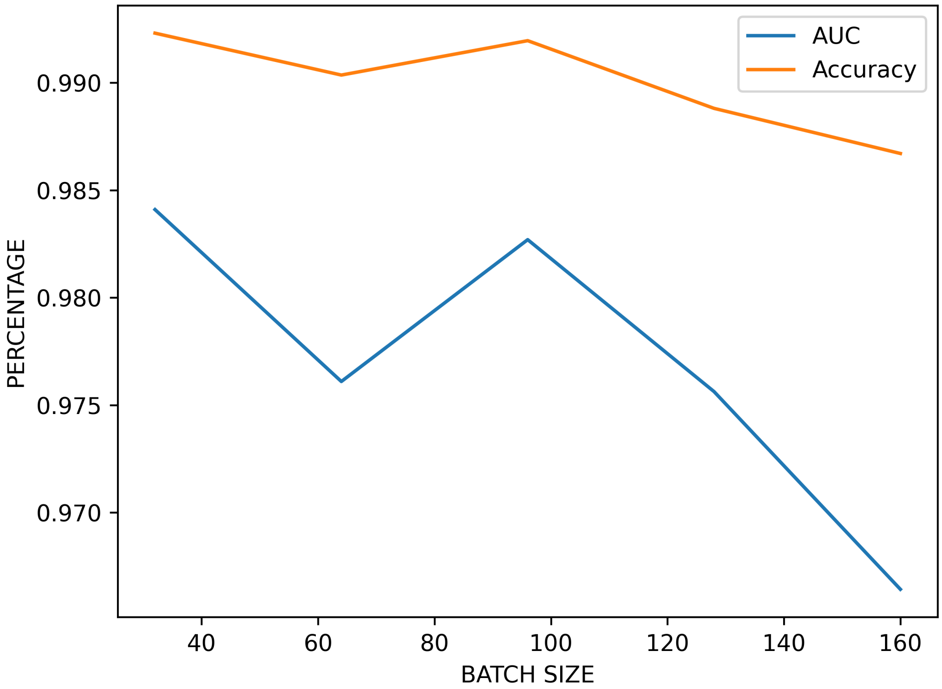

FIGURE 3 Different Batch Size

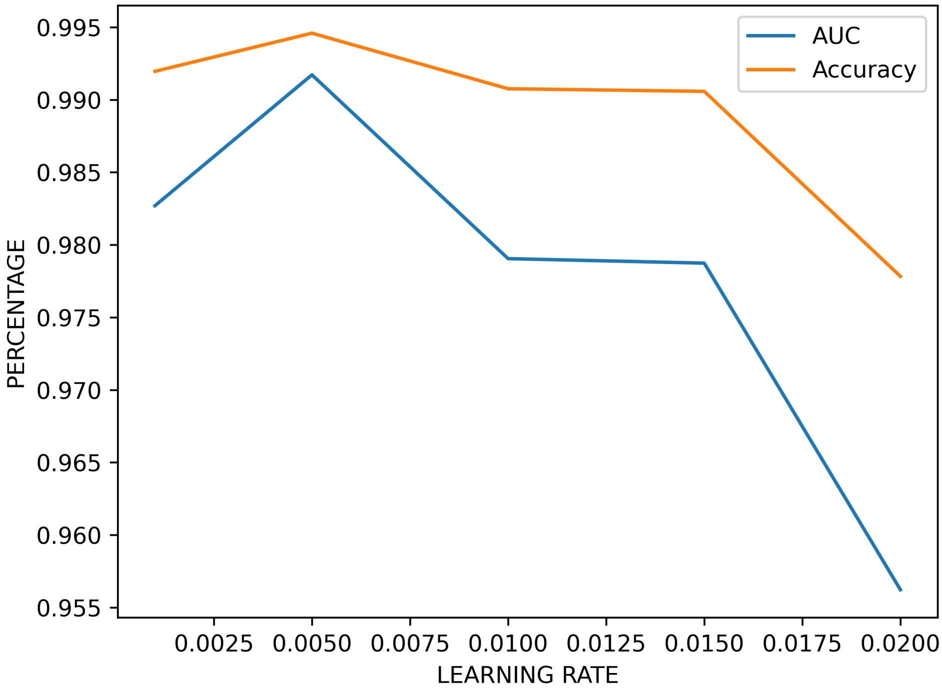

FIGURE 4 Different Learning Rate

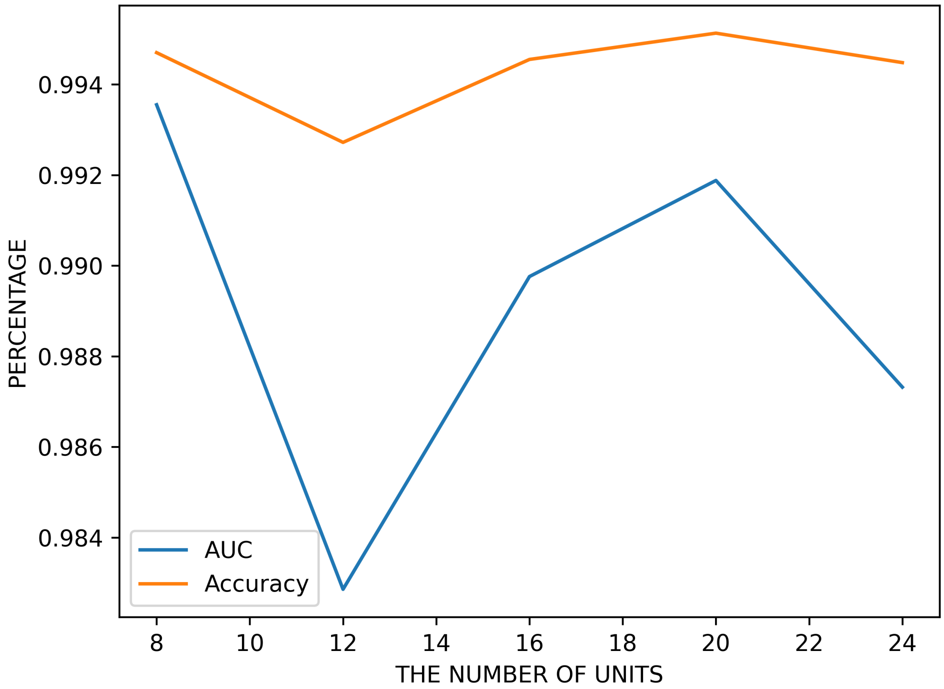

FIGURE 5 Different Number of Units in GRU

meaning 80% of the data is for training and 20% is for validation or testing. This helps to evaluate the model's performance on unseen data and avoid overfitting or underfitting.

5.2. **Comparison.** In this part, we compare the performance of the proposed model with four well-known deep learning models, namely LSTM [22], GRU, SVM [23], and CNN [24]. GRU model has been introduced in section 4.4. LSTM is another type of RNN Like GRU that can process data sequentially and learn long-term dependencies. It is designed to overcome the vanishing gradient problem that occurs in traditional RNNs, which makes them unable to learn long-term dependencies. LSTM also has a special structure that consists of a cell, an input gate, an output gate, and a forget gate. The cell stores information over long periods, and the gates control how much information to keep, discard, or output from the cell. An SVM model can perform classification, regression, and outlier detection tasks. It works by finding the optimal hyperplane that separates the data points into different classes or predicts their values. The hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the closest data points of each class. The data points that lie on the margin are called support vectors, and they are the only ones that affect the hyperplane. SVM models can handle high-dimensional and nonlinear data by using different kernel functions, which transform the data into a higher-dimensional space where the hyperplane can be found. A CNN model can process data with a grid-like structure, such as images, audio, or text. A CNN model consists of multiple layers, each of which performs a specific operation on the input data. The main types of layers in a CNN model are the convolutional layer, pooling layer and fully connected layer. A CNN model can have multiple convolutional and pooling layers, stacked on top of each other, forming a deep network that can learn complex and hierarchical features from the data.

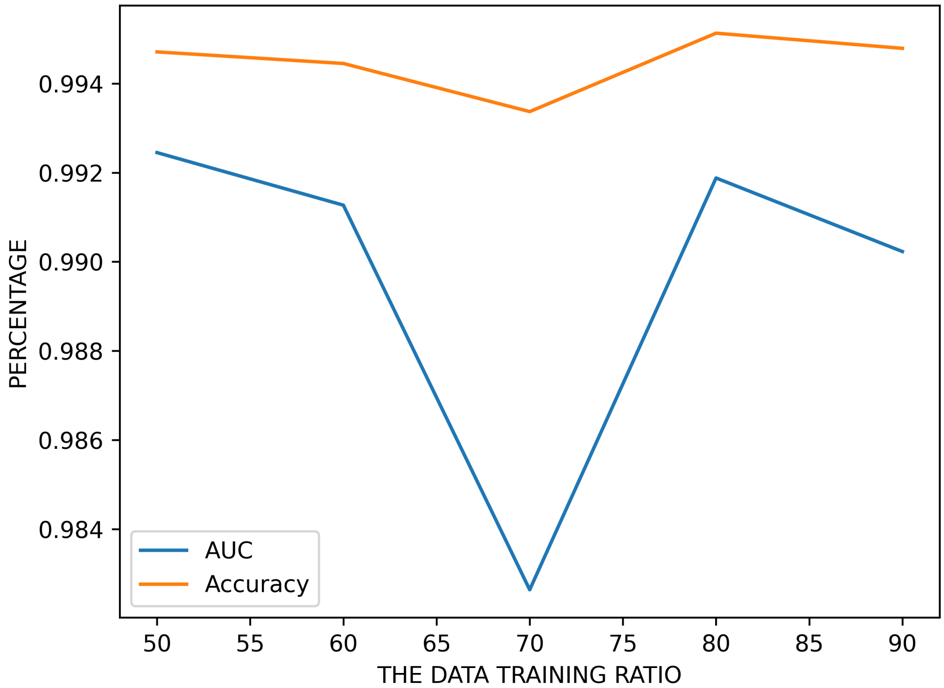

FIGURE 6 Different Training Ratio

TABLE 2 COMPARISON WITH OTHER MODELS

Models	Accuracy(%)	AUC(%)	Number of Params
Ours	**99.48**	**98.94**	**1592**
LSTM	99.23	98.13	2341
GRU	99.31	98.22	1821
SVM	97.21	94.81	-
CNN	98.18	95.96	1801

We choose a comparative model size level for these models. Specifically, the number of parameters for each model is quite similar. Besides, we use the same training ratio for each model, namely the same proportion of data for training and testing. The performance of each model is evaluated by two metrics, namely prediction accuracy and Area Under the ROC Curve (AUC). As is shown in Table 2, the proposed model outperforms the other four models both in prediction accuracy and AUC.

6. **Conclusion.** In this paper, we have introduced the "LW-Risk Prediction" model, an innovative deep-learning solution aimed at improving the prediction of risks in the challenging domain of high-voltage direct current (DC) transmission line live working. As demonstrated, live working risk prediction on DC transmission lines presents two significant challenges: the dynamic nature of risk factors which are continually changing over time; and necessitating timely risk predictions. Our exploration of the "LW-Risk Prediction" model's design elements has revealed its pioneering approach to these challenges. By incorporating domain-specific expertise, the model adeptly handles the intricate nuances of DC transmission line environments. The utilization of recurrent neural networks (RNNs) with attention mechanisms allows the model to capture the temporal dynamics

of risk factors, thus addressing the first challenge. Furthermore, the model effectively integrates both categorical and numerical data to offer a comprehensive risk assessment. Additionally, the inclusion of an anomaly factor provides additional safety, identifying unusual patterns that may indicate heightened risk. The experimental results attest to the model's effectiveness in addressing these challenges. LW-Risk Prediction demonstrates its capacity to harness diverse, time-varying risk factors, delivering accurate and timely predictions for DC transmission line live working risk. As we continue to refine and expand this innovative model, it stands as a testament to the power of creative problem-solving and advanced technology in safeguarding the well-being of professionals in the field of electrical safety.

REFERENCES

[1] P. Martini, *Live-line working and evaluation of risk on 400kV transmission line.* The University of Manchester (United Kingdom), 2017.

[2] P. Yong, L. Kai, W. Tian, H. Zehua, C. Zhang, and Z. Yadi, "Research on safety protection of±800kv uhv dc line live working," in *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA).* IEEE, 2017, pp. 958–961.

[3] Y. Hu, K. Liu, T. Wu, T. Liu, B. Xiao, Y. Peng, and Z. Su, "Key technology research and application of live working technology on ehv/uhv transmission lines in china," in *2014 International Conference on Power System Technology.* IEEE, 2014, pp. 2299–2309.

[4] M. Zhenyu, C. Chao, G. Leijiao, W. Binbin, Z. Jinhui, and W. Ziling, "Feasibility analysis and protection measures of live working on the multi-circuit transmission lines on the same tower," in *Journal of Physics: Conference Series*, vol. 1550, no. 5. IOP Publishing, 2020, p. 052004.

[5] International Electrotechnical Commission, *Live working - Minimum approach distances for a.c. systems in the voltage range 72,5 kV to 800 kV - A method of calculation*, IEC Std. 61 472, 2013. [Online]. Available: https://webstore.iec.ch/publication/5486

[6] IEEE, *IEEE Guide for Maintenance Methods on Energized Power Lines*, IEEE Std. 516-1995, 1995. [Online]. Available: https://standards.ieee.org/standard/516-1995.html

[7] S. Jiang, P. Kuffel, and R. Adapa, "Live work insulation considerations on hvdc lines," in *2020 IEEE Electric Power and Energy Conference (EPEC).* IEEE, 2020, pp. 1–5.

[8] J. Gao, L. Wang, S. Wu, K. Liu, Y. Peng, J. Liu, and Y. Liu, "Prediction of breakdown voltage of equipotential live-line work air gaps in transmission lines: A computational discharge model," *IEEE Transactions on Electromagnetic Compatibility*, vol. 64, no. 6, pp. 1932–1940, 2022.

[9] W. Tian, Z. Liyun, H. Shuai, Y. Dong, and C. Cong, "Equal-potential entering path and arc energy calculation method for live working on±800kv dc transmission line based on fem," in *2019 4th International Conference on Intelligent Green Building and Smart Grid (IGBSG).* IEEE, 2019, pp. 631–638.

[10] C. Borgelt, "Frequent item set mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 6, pp. 437–456, 2012.

[11] S. Xingguo, Y. Qiang, X. Zhiliang, A. Shan, and Z. Chaoqun, "High voltage transmission line live working risk early warning method based on big data frequent itemset mining," 2022.

[12] W.-S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie, "Fault tree analysis, methods, and applications : a review," *IEEE Transactions on Reliability*, vol. 34, no. 3, pp. 194–203, 1985.

[13] J. D. Andrews and S. J. Dunnett, "Event-tree analysis using binary decision diagrams," *IEEE Transactions on Reliability*, vol. 49, no. 2, pp. 230–238, 2000.

[14] I. Ben-Gal, "Bayesian networks," *Encyclopedia of statistics in quality and reliability*, 2008.

[15] L. A. Zadeh, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, 1988.

[16] Z. Qiu, J. Ruan, C. Liu, Q. Jin, and D. Huang, "Discharge voltage prediction of complex gaps for helicopter live-line work: An approach and its application," *Electric Power Systems Research*, vol. 164, pp. 139–148, 2018.

[17] J. Guo, Q. Wu, K. Guo, S. Xiong, W. Feng, and J. Xue, "Study on the construction and application of digital twins on high voltage transmission line live working scenes," *IEEE Access*, vol. 9, pp. 111 587–111 594, 2021.

[18] "DC transmission line risk factors," 2023. [Online]. Available: https://github.com/DataFreeShare/Lwrisk.git

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[20] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, no. 64-67, p. 2, 2001.

[21] A. Turpin and F. Scholer, "User performance versus precision measures for simple search tasks," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 11–18.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] M. Biswal and P. K. Dash, "Measurement and classification of simultaneous power signal patterns with an s-transform variant and fuzzy decision tree," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 1819–1827, 2012.

[24] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2017.