# Comparative Analysis of SMOTE-Enhanced Deep Learning and Machine Learning Models on a SCADA Dataset

1st Esra SÖĞÜT
*Department of Computer Engineering*
*Faculty Of Technology*
*Gazi University*
Ankara, Türkiye
0000-0002-0051-2271

2nd Aynur KOÇAK
*Department of Electrical and Electronics Engineering*
*Faculty Of Technology*
*Gazi University*
Ankara, Türkiye
0000-0001-9647-7281

3rd O. Ayhan ERDEM
*Department of Computer Engineering*
*Faculty Of Technology*
*Gazi University*
Ankara, Türkiye
0000-0001-7761-1078

2025 18th International Conference on Information Security and Cryptology (ISCTürkiye) | 979-8-3315-5709-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISCTRKIYE68593.2025.11224853

***Abstract*— This study aims to classify the "specific result" output, a rarely explored aspect in the literature, using a widely adopted dataset in SCADA system security. While most previous studies have concentrated on binary or categorized result outputs, this research uniquely focuses on specific result classification. To mitigate class imbalance and enhance model performance, the Synthetic Minority Oversampling Technique (SMOTE) was applied. Subsequently, multiple machine learning and deep learning algorithms, including Random Forest, Logistic Regression, Decision Tree, Naive Bayes, Support Vector Machine, Convolutional Neural Network, Long Short-Term Memory, Extended Long Short-Term Memory, Multilayer Perceptron, Deep Neural Network, and Gated Recurrent Unit, were implemented. Experimental results demonstrated that the Random Forest model achieved an accuracy of 99%, while the MLP model reached 95%. These results outperform previous studies on specific result classification and confirm the effectiveness of the proposed approach for attack detection in Industrial Control Systems.**

**Keywords— Multi-class; Industrial Control System; Random Forest; MLP; specific result; cyber security**

## I. INTRODUCTION

Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) systems play a central role in ensuring the secure and efficient operation of critical infrastructures such as energy, water, transportation, and manufacturing. With technological advancements such as Industry 4.0 and the Internet of Things (IoT), these systems have become network-connected and remotely accessible, which enhances operational efficiency but also introduces new security risks [1]. Since ICS and SCADA systems are designed to maintain the continuity of industrial processes, any cyberattack can lead to the interruption of these processes, economic losses, and even severe consequences that may endanger human safety. In recent years, the number and complexity of cyberattacks targeting ICS and SCADA systems have increased. These attacks include a wide range of threats such as Denial-of-Service (DoS/DDoS), command injection, data manipulation, and unauthorized access. The literature presents numerous Machine Learning (ML)-based and Deep Learning (DL)-based methods developed to detect such threats. However, the vast majority of these studies generally focus on less detailed output types, such as binary results (attack/normal) [2,3].

In this study, the targeted output is a "specific result" classification, which is tailored to the dataset used and rarely addressed in the literature. The "specific result" classification enables the detection of attacks at a more detailed level, thereby contributing to the implementation of proactive measures in ICS security. The dataset employed in this study consists of both attack instances and normal traffic samples collected from a SCADA environment. To address class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied. This method enhanced the representation of minority classes, thereby improving the model's ability to recognize rare attack types.

During the modeling process, both ML and DL algorithms were utilized. Among the ML algorithms, Random Forest (RF), Logistic Regression (LR), Decision Tree (DT), Naive Bayes (NB), and Support Vector Machines (SVM) were employed, while the DL models evaluated included Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Extended Long Short-Term Memory (XLSTM), Multilayer Perceptron (MLP), Deep Neural Network (DNN), and Gated Recurrent Unit (GRU). The results revealed that Random Forest (99% accuracy) in the ML category and MLP (95% accuracy) in the DL category achieved the highest performance in "specific result" classification. In this context, the study not only addresses a gap in the literature but also provides a strong reference for detailed attack detection in ICS security.

## II. RELEATED WORKS

In the study by Morris et al., the generated dataset was compared against a physical model of a virtual pipeline, evaluating Normal, Startup, and Attack operations [4]. In a subsequent work, a misuse-based intrusion detection system was developed for SCADA systems in gas pipeline infrastructure. The authors enhanced classification performance by performing feature extraction on the existing dataset, achieving binary classification accuracies of 99.97% (RF) and 99.91% (PART) using NB, PART, and RF algorithms [5]. Within the scope of ICS security, ML and DL models have been applied for anomaly detection. In [6], preprocessing steps involved removing missing values, applying label encoding, and addressing class imbalance through SMOTE. Models including NB, PART, Random Tree (RT), and MLP-PART were tested, with RT achieving 99.70% and PART achieving 94.14% accuracy. When a model is trained directly on an imbalanced dataset, new malicious data may be misclassified. To address this, the authors proposed an ensemble deep representation learning model based on SAE, aimed at improving overall model

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

Authorized licensed use limited to: ULAKBIM UASL - GAZI UNIV. Downloaded on November 27,2025 at 11:39:01 UTC from IEEE Xplore. Restrictions apply.

performance. The proposed method was compared with DT, DNN, and RF, and achieved a superior accuracy rate of 96.00% [2].

Another study introduced a method for cyberattack detection in ICS environments that integrates Autoencoder, DCNN, and LSTM-based DL models in different configurations. The objective was to develop a robust and generalizable solution capable of detecting both known and zero-day attacks with high accuracy. The DCNN component automatically extracted high-level features from the dataset, the LSTM captured temporal dependencies and sequential patterns of attacks, and the Autoencoder contributed by denoising and performing dimensionality reduction for more efficient feature representation. In this study, the DCNN model achieved an accuracy of 69.00% [3]. To capture both spatial and temporal features of network traffic, another work proposed a hybrid DL intrusion detection model combining CNN and BiLSTM. During preprocessing, the SMOTE-ENN (Synthetic Minority Oversampling Technique – Edited Nearest Neighbors) algorithm was applied to address data imbalance and improve the detection of minority classes [7]. In this CNN-BiLSTM approach presented by Wang et al., accuracy rates of 85.84% and 85.50% were reported.

For improved ICS cyberattack detection, a method based on 1D-CNN and Wasserstein Generative Adversarial Networks (WGAN) was introduced. WGAN was employed to address data imbalance and generate synthetic samples for minority classes, followed by feature extraction from time-series network traffic using a 1D-CNN architecture. The 1D CWGAN-BiSRU model achieved an accuracy of 99.00% [8].

Another study highlighted that existing industrial intrusion detection methods fail to adequately account for the specific characteristics of industrial control flows. To address this limitation, a novel detection algorithm combining the Improved Gray Wolf Optimizer (IGWO) and Gated Recurrent Unit (GRU) was proposed, achieving a categorized classification accuracy of 97.62% [9].

Finally, a new technique focused on detecting malicious activities and predicting future threats by combining Marked Temporal Point Processes (MTTP) with neural networks. Evaluations using the dataset were conducted with both RMTPP and ERPP models, where the ERPP achieved 95.50% (categorized – Precision) and 93.50% (specific – Precision) accuracy [10].

## III. Methodology

This section presents the methodological framework of the study, covering dataset description, preprocessing steps, class imbalance handling, and the application of both ML and DL algorithms. The approach also includes the use of the "specific result" classification scheme, which targets the identification of 35 distinct attack types in the dataset. Furthermore, performance evaluation metrics and experimental settings are detailed to ensure the reproducibility and clarity of the proposed method.

### *A. Dataset*

The dataset employed in this study is the New Gas Pipeline Dataset, introduced by Morris et al. (2015) at the 7th Annual Southeastern Cyber Security Summit under the title "Industrial Control System Simulation and Data Logging for Intrusion Detection System Research" [4]. This dataset is widely used in SCADA and critical infrastructure cybersecurity research [2-10] and was developed to support studies on ICS security.

It comprises two main components:

- Physical system: a laboratory-scale replica of a real gas pipeline, operated under both normal conditions and 35 distinct cyberattack scenarios.
- Virtual pipeline system: incorporating HMI, PLC, processor, and network components communicating via the Modbus/TCP protocol, generating realistic network traffic.

The dataset contains a total of 274,628 instances, including 214,580 normal Modbus network packets and 60,048 attack packets. Labels are assigned as "0" for normal traffic and "1–35" for different attack types, with per-class sample sizes ranging from 666 to 2,204. Features include both numerical and categorical attributes, making the dataset suitable for developing ML and DL-based intrusion detection systems.

While previous studies using this dataset have predominantly focused on binary classification (attack/normal) or attack category classification, the present work differs by addressing the specific result setting, performing experiments on the classification of all 35 individual attack types.

### *B. Dataset Preprocessing*

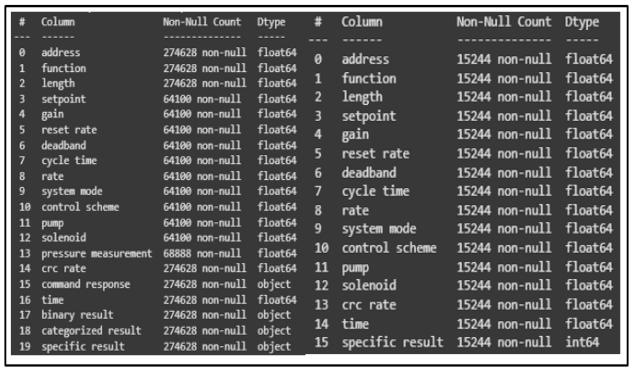
In this study, the "specific result" class of the dataset is targeted, with experiments conducted for 35 distinct attack types. Prior to preprocessing, the dataset contained 274,628 instances. The first step involved applying label encoding to the "specific result" column, assigning unique labels in the range of 0–35. Rows with missing values were removed, along with columns containing excessive missing data or irrelevant outputs (e.g., command response, binary result, categorized result).

As the dataset was highly imbalanced, with label "0" (normal state) comprising 214,580 samples compared to significantly fewer samples for attack classes (e.g., 2,204, 2,176, 1,160), all "0" label instances were removed to exclude normal operation from classification. This adjustment allowed the focus to remain solely on classifying attack types. However, imbalance among attack classes persisted. To address this, SMOTE was applied, generating synthetic samples for minority classes and creating a more balanced class distribution.

Subsequently, the dataset was divided into 70% training and 30% testing, with 20% of the training portion reserved for validation during model training. The characteristics of the dataset before and after preprocessing are presented in Figure 1.

The preprocessing steps involved cleaning the data by handling missing values, converting categorical variables into numerical formats, and standardizing the dataset. During preprocessing, both the number of columns and rows were reduced, and the attack categories in the "specific result" column were decreasing from 35 distinct types to 18. As a result, the dataset became more compact and consistent, facilitating efficient model training and evaluation.

Authorized licensed use limited to: ULAKBIM UASL - GAZI UNIV. Downloaded on November 27,2025 at 11:39:01 UTC from IEEE Xplore. Restrictions apply.



| # | Column | Non-Null Count | Dtype |
|---|---|---|---|
| 0 | address | 274628 non-null | float64 |
| 1 | function | 274628 non-null | float64 |
| 2 | length | 274628 non-null | float64 |
| 3 | setpoint | 64100 non-null | float64 |
| 4 | gain | 64100 non-null | float64 |
| 5 | reset rate | 64100 non-null | float64 |
| 6 | deadband | 64100 non-null | float64 |
| 7 | cycle time | 64100 non-null | float64 |
| 8 | rate | 64100 non-null | float64 |
| 9 | system mode | 64100 non-null | float64 |
| 10 | control scheme | 64100 non-null | float64 |
| 11 | pump | 64100 non-null | float64 |
| 12 | solenoid | 64100 non-null | float64 |
| 13 | pressure measurement | 68888 non-null | float64 |
| 14 | crc rate | 274628 non-null | float64 |
| 15 | command response | 274628 non-null | object |
| 16 | time | 274628 non-null | float64 |
| 17 | binary result | 274628 non-null | object |
| 18 | categorized result | 274628 non-null | object |
| 19 | specific result | 274628 non-null | object |

| # | Column | Non-Null Count | Dtype |
|---|---|---|---|
| 0 | address | 15244 non-null | float64 |
| 1 | function | 15244 non-null | float64 |
| 2 | length | 15244 non-null | float64 |
| 3 | setpoint | 15244 non-null | float64 |
| 4 | gain | 15244 non-null | float64 |
| 5 | reset rate | 15244 non-null | float64 |
| 6 | deadband | 15244 non-null | float64 |
| 7 | cycle time | 15244 non-null | float64 |
| 8 | rate | 15244 non-null | float64 |
| 9 | system mode | 15244 non-null | float64 |
| 10 | control scheme | 15244 non-null | float64 |
| 11 | pump | 15244 non-null | float64 |
| 12 | solenoid | 15244 non-null | float64 |
| 13 | crc rate | 15244 non-null | float64 |
| 14 | time | 15244 non-null | float64 |
| 15 | specific result | 15244 non-null | int64 |

*Figure 1 Dataset Features Before and After Data Preprocessing*

### C. Performance Metrics

In order to comprehensively evaluate the classification performance of the proposed models, multiple performance metrics were employed. These metrics provide insights into both overall accuracy and class-specific detection capabilities, particularly in the presence of class imbalance. The evaluation criteria are as follows:

- Accuracy: The ratio of correctly classified instances to the total number of instances; used to measure the overall success rate.
- Macro Precision: The average of precision values calculated for all classes, treating each class with equal weight.
- Macro Recall: The average recall across all classes; unaffected by class imbalance.
- Macro F1-Score: The harmonic mean of Macro Precision and Macro Recall.
- Weighted Precision: The precision values averaged with weights proportional to the number of samples in each class.
- Weighted Recall: The recall values averaged with weights proportional to the number of samples in each class.
- Weighted F1-Score: The harmonic mean of Weighted Precision and Weighted Recall.
- Confusion Matrix: A matrix visualizing the number of correct and incorrect classifications for each class; used to identify misclassifications between classes.

## IV. Experiments and Results

A total of 11 models from both ML and DL algorithms were employed for data analysis. Model selection was based on existing studies in the literature. According to the obtained results, RF achieved the highest performance among ML methods, while MLP achieved the highest performance among DL methods.

*Table 1 Comparison of All Metrics of Models*

| Model | Category | Accuracy | Macro Precision | Macro Recall | Macro F1 | Weighted Precision | Weighted Recall | Weighted F1 |
|---|---|---|---|---|---|---|---|---|
| RF | ML | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| LR | | 0.11 | 0.07 | 0.11 | 0.06 | 0.08 | 0.11 | 0.07 |
| DT | | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 |
| NB | | 0.347 | 0.3638 | 0.3248 | 0.2797 | 0.3825 | 0.347 | 0.2982 |
| SVM | | 0.9366 | 0.9335 | 0.9348 | 0.9334 | 0.9373 | 0.9366 | 0.9362 |
| CNN | DL | 0.9027 | 0.9015 | 0.9036 | 0.8979 | 0.9082 | 0.9027 | 0.9008 |
| LSTM | | 0.5551 | 0.5754 | 0.5378 | 0.4906 | 0.5872 | 0.5551 | 0.5093 |
| XLSTM | | 0.1032 | 0.0221 | 0.0873 | 0.0311 | 0.0257 | 0.1032 | 0.0367 |
| MLP | | 0.9526 | 0.95 | 0.9509 | 0.9498 | 0.9527 | 0.9526 | 0.9521 |
| DNN | | 0.9276 | 0.9255 | 0.9267 | 0.9246 | 0.9286 | 0.9276 | 0.9268 |
| GRU | | 0.0715 | 0.004 | 0.0556 | 0.0074 | 0.0051 | 0.0715 | 0.0095 |

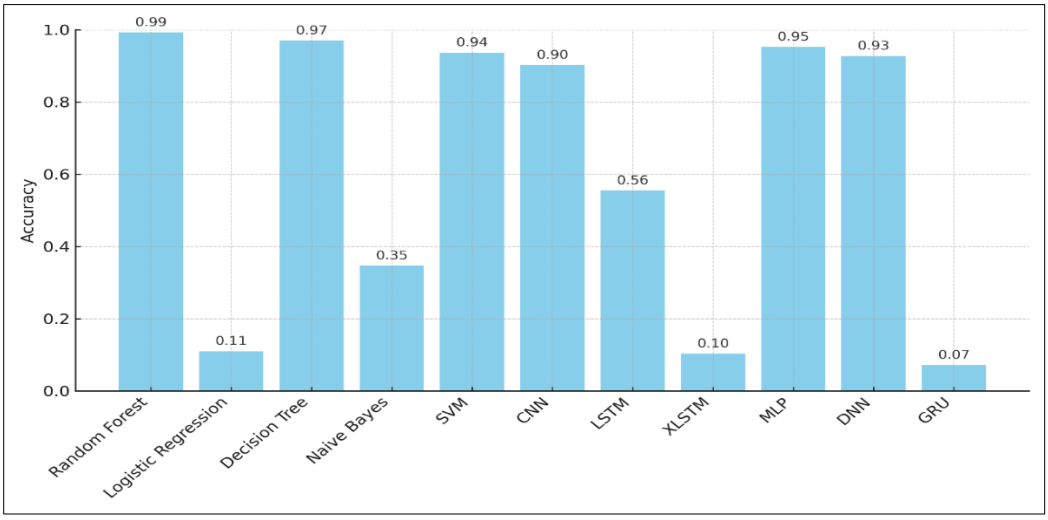



*Figure 2 Accuracy Comparison of Models*

Authorized licensed use limited to: ULAKBIM UASL - GAZI UNIV. Downloaded on November 27,2025 at 11:39:01 UTC from IEEE Xplore. Restrictions apply.

The models used in the experiments were categorized into ML and DL approaches (Table 1). Based on the average performance values, tree-based methods such as RF and DT outperformed DL models. Within the DL category, fully connected architectures (MLP, DNN) achieved better results compared to more complex models such as CNN, LSTM, XLSTM, and GRU. Considering the highest accuracy scores, RF (0.99), DT (0.97), and MLP (0.95) ranked as the top three models (Figure 2). In contrast, the lowest-performing models were NB (0.35), LR (0.11), XLSTM (0.10), and GRU (0.07).

A comparative overview of related studies utilizing the same dataset and various ML and DL approaches is presented in Table 2, providing a contextual benchmark for interpreting the experimental results of this study.

*Table 2 Literature Summary of Classification Approaches*

| Ref. | Model | Classification | Accuracy |
|---|---|---|---|
| [5] | NB, PART, RF | Binary result categorized result | %99,97 |
| [6] | NB, PART, RT, MLP PART | Binary result categorized result | %99,70 |
| [2] | Proposed, DT, DNN, RF | Binary result | %96.00 |
| [3] | DCNN, LSTM | Binary result | %69.00 |
| [7] | CNN-BiLSTM | Binary result categorized result | %85.84 |
| [8] | GAN-CNN, GAN-BiSRU, 1D CWGAN-CNN, 1D CWGAN-BiSRU | categorized result | %99.00 |
| [9] | IGWO-GRU | categorized result | %97.62 |
| [10] | RMTPP, ERPP | categorized result specific result | %95.50 |

In contrast to most previous studies, which primarily addressed binary or categorized classification tasks, this study focuses on the more challenging specific result classification. In our experiments, RF and DT achieved accuracy values of 99% and 97%, respectively, which are in line with previously reported results using similar models [5,6,9]. The MLP model also reached 95.26% accuracy, showing consistency with other DL-based approaches in the literature [2,10]. On the other hand, sequence-based models such as LSTM and GRU provided lower accuracy, which is comparable to earlier findings [3]. These outcomes suggest that, despite the increased complexity of specific result classification, our approach produces results that are broadly consistent with trends observed in prior research.

Figure 3 clearly illustrates the prediction performance of the RF model across different classes. The model demonstrates both high accuracy and strong generalization capability in a multi-class setting. Only minor misclassifications (1–4 instances) are observed in certain classes, typically occurring between classes with similar characteristics. The low number of misclassifications also contributes to maintaining high precision and recall values.

The MLP confusion matrix (Figure 4), while generally achieving high accuracy, reveals a slightly higher error rate in some classes compared to RF. For example, there are minimal cross-errors between Class 1 and Class 2, whereas more noticeable confusion is observed in certain other classes (e.g., Class 31 and Class 32).

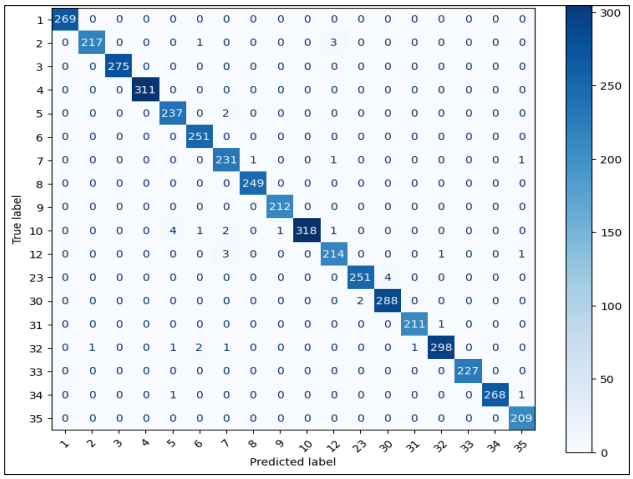



*Figure 3 Random Forest Confusion Matrix*

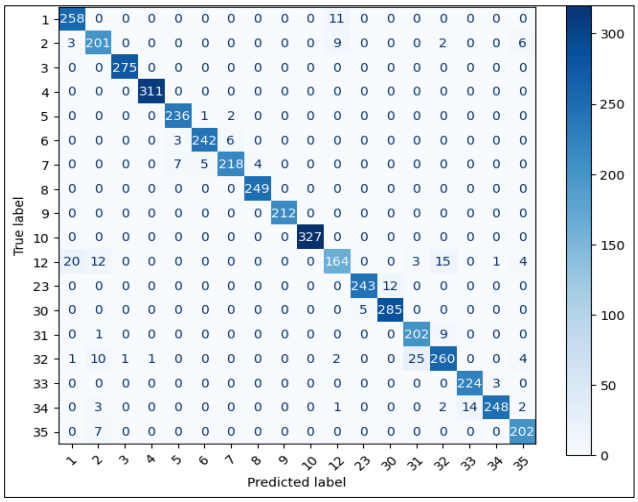



*Figure 4 MLP Confusion Matrix*

The errors are predominantly concentrated among a few neighboring classes, indicating that the models encounter challenges in distinguishing between classes with similar characteristics. Compared to the RF confusion matrix, the MLP model exhibits a slightly higher misclassification rate. While SMOTE helps mitigate class imbalance and can improve correct classification rates for underrepresented classes, the synthetic samples it generates may occasionally complicate the model's decision boundaries, leading to increased error rates in certain classes.

## V. Conclusion

In this study, the classification of the specific result output, which has been rarely addressed in the literature, was performed using a widely adopted dataset in SCADA system cybersecurity. To this end, various ML and DL algorithms were applied to detect multiple attack types. Since the dataset suffered from class imbalance, the SMOTE technique was employed, which improved model performance, particularly for minority classes. Importantly, focusing on the specific result output provides practical value for real-world SCADA/ICS environments, as it enables not only the

Authorized licensed use limited to: ULAKBIM UASL - GAZI UNIV. Downloaded on November 27,2025 at 11:39:01 UTC from IEEE Xplore. Restrictions apply.

detection of an attack but also the identification of its type, thereby supporting faster and more effective incident response. The experimental results demonstrated that traditional ML methods, such as RF and DT, achieved high accuracy and F1-scores, with the RF model standing out with an accuracy rate of 99%. In this study, the accuracy values of GRU, LSTM, CNN, and LR algorithms were found to be significantly lower than other methods. One of the main reasons for this is that the data used lacks a sequential structure, and therefore, it cannot effectively utilize the inherent advantages of recurrent neural networks and convolutional architectures. LR, on the other hand, cannot adequately represent complex patterns in high-dimensional and multi-class data because it can only learn linear decision boundaries. Furthermore, class imbalance and the multi-class problem structure negatively impacted the generalization ability of these models. The findings indicate that methods such as tree-based algorithms and multi-layer artificial neural networks are more suitable and successful when applied to tabular feature vectors.

For future work, techniques such as hyperparameter optimization and the development of ensemble methods can be employed to further improve model performance. Additionally, utilizing different ICS datasets and applying various data augmentation techniques could enhance the generalization capability of the models.

## REFERENCES


[1] Polat, H., Türkoğlu, M., Polat, O., & Şengür, A. A novel approach for accurate detection of the DDoS attacks in SDN-based SCADA systems based on deep recurrent neural networks. *Expert Systems with Applications*, vol. 197, pp. 116748, 2022.

[2] Al-Abassi, A., Karimipour, H., Dehghantanha, A., & Parizi, R. M. An ensemble deep learning-based cyber-attack detection in industrial control system. *Ieee Access*, vol. 8, pp. 83965-83973, 2020.

[3] Saheed, Y. K., Misra, S., & Chockalingam, S. Autoencoder via DCNN and LSTM models for intrusion detection in industrial control systems of critical infrastructures. In: *2023 IEEE/ACM 4th International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS)*. IEEE, pp. 9-16, May 2023.

[4] Morris, T. H., Thornton, Z., & Turnipseed, I. Industrial control system simulation and data logging for intrusion detection system research. *7th annual southeastern cyber security summit*, pp. 1-6, 2015.

[5] Khan, A. A. Z. Misuse intrusion detection using machine learning for gas pipeline SCADA networks. In: *Proceedings of the international conference on security and management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), pp. 84-90, 2019.

[6] Turnipseed, I. P. A new SCADA dataset for intrusion detection system research, A Thesis, Faculty of Mississippi State University, the Degree of Master of Science, Electrical and Computer Engineering, August 2015.

[7] Wang, J., Si, C., Wang, Z., & Fu, Q. A New Industrial Intrusion Detection Method Based on CNN-BiLSTM. *Computers, Materials & Continua*, vol. 79(3), 2024.

[8] Cai, Z., Du, H., Wang, H., Zhang, J., Si, Y., & Li, P. One-Dimensional Convolutional Wasserstein Generative Adversarial Network Based Intrusion Detection Method for Industrial Control Systems. *Electronics,* vol. 12(22), 4653, 2023.

[9] Yang, W., Shan, Y., Wang, J., & Yao, Y. An industrial network intrusion detection algorithm based on IGWO-GRU. *Cluster Computing*, vol. 27(6), pp. 7199-7217, 2024.

[10] Fortino, G., Greco, C., Guzzo, A., & Ianni, M. Identification and prediction of attacks to industrial control systems using temporal point processes. *Journal of Ambient Intelligence and Humanized Computing*, vol. 14(5), pp. 4771–4783, 2023.


Authorized licensed use limited to: ULAKBIM UASL - GAZI UNIV. Downloaded on November 27,2025 at 11:39:01 UTC from IEEE Xplore. Restrictions apply.