Anomaly Detection In Power Distribution Networks: A Comparative Study Of Classifiers

Mr. P Satish[1*], G. Gnana prasuna[2], Manoj kumar[2], Y. Prashanth[2], G. Abhinav[2]

[1,2]Department of Computer Science and Engineering(Cyber Security), Sree Dattha Group of Institutions, Sheriguda, Hyderabad, Telangana.

*Corresponding author: Mr. P. Satish

ABSTRACT

Machine learning (ML)-based classification techniques offer promising solutions for identifying failures in Intelligent Electronic Devices (IEDs) within smart power grid systems. These systems are integral components of modern power grids, facilitating efficient energy management and ensuring reliable electricity supply. However, the complexity and interconnected nature of smart grid infrastructures make them susceptible to various failures and attacks, necessitating robust fault detection mechanisms. Another crucial application lies in cybersecurity for smart grids, where ML algorithms can aid in detecting and mitigating attacks targeting IEDs. By analyzing network traffic patterns and abnormal behaviors in IEDs, ML models can identify suspicious activities indicative of failures, such as unauthorized access attempts or tampering with device configurations. This proactive approach to cybersecurity enhances the resilience of smart grid systems against malicious threats, safeguarding critical infrastructure and ensuring uninterrupted electricity supply to consumers. Current methods for detecting IED failures in smart power grid systems often rely on rule-based approaches or manual inspection, which are labor-intensive and prone to errors. These traditional techniques may overlook subtle patterns or anomalies indicative of emerging failures, leading to delayed responses and increased risk of system downtime. Additionally, existing fault detection mechanisms may struggle to differentiate between genuine failures and benign fluctuations in system behavior, resulting in false alarms and unnecessary maintenance interventions. To address the limitations of existing fault detection methods, this work proposes a novel ML-based classification system for identifying failures in IEDs within smart power grid systems. The proposed system leverages supervised learning algorithms trained on labeled datasets derived from power system attack scenarios. By analyzing various features extracted from IED data, such as voltage fluctuations, current readings, and communication patterns, our ML models can accurately classify different types of failures, including equipment malfunctions, failures, and operational errors.

Keywords: Power Distribution Networks, Smart Grid, Intelligent Electronic Devices (IEDs), Fault Detection, Smart Power Grid Systems.

1. INTRODUCTION

In the realm of modern power grids, Intelligent Electronic Devices (IEDs) play a pivotal role in ensuring efficient energy management and reliable electricity supply. However, the intricate nature of smart grid infrastructures renders them susceptible to diverse failures and potential cyberattacks. Traditional methods of detecting IED failures often rely on rule-based approaches or manual inspection, which are labor-intensive and prone to errors. These methods may miss subtle patterns indicative of emerging failures, leading to delayed responses and heightened risks of system downtime. To address these shortcomings, this work proposes a novel approach: leveraging machine learning (ML)-based classification techniques to proactively identify failures in IEDs within smart power grid systems. By employing supervised learning algorithms trained on labeled datasets derived

from power system attack scenarios, our ML models can accurately classify various types of failures, including equipment malfunctions, failures, and operational errors. This proactive approach enhances the resilience of smart grid systems against both failures and malicious threats, ensuring uninterrupted electricity supply to consumers.

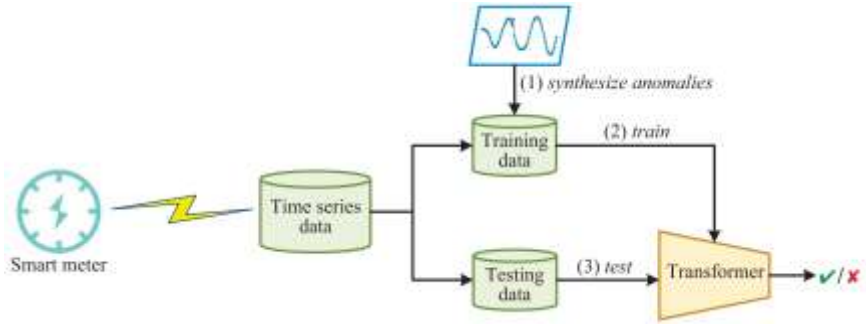

Fig. 1: Detecting Anomaly in Power Distribution.

The complexity and interconnected nature of smart grid infrastructures pose significant challenges in effectively detecting and mitigating failures in Intelligent Electronic Devices (IEDs). Existing methods for identifying IED failures, such as rule-based approaches or manual inspection, are often labor-intensive, error-prone, and may overlook subtle patterns indicative of emerging failures. Moreover, distinguishing between genuine failures and benign fluctuations in system behavior remains a challenge, leading to false alarms and unnecessary maintenance interventions. Thus, there is a critical need for robust fault detection mechanisms that can accurately classify different types of failures in IEDs, enabling proactive maintenance and enhancing the resilience of smart grid systems against both operational errors and malicious attacks.

2. LITERATURE SURVEY

Condition Monitoring (CM) of electrical power grids is an essential anomaly prevention process enabling higher quality continuous delivery of electrical energy with hopefully zero downtimes [1, 2]. Today's advanced computing and networking technologies make power grids CM more ergonomic and an accessible centralized process led by the so-called Internet of Things (IoT) technologies [3], [4], [5]. SGs are a combination of two interconnected layers, in particular, a cyber-layer and a physical one [6]. The cyber layer is a blend of computers networking technologies and necessary monitoring applications and software. The physical layer compromises physical elements and field devices (e.g., smart sensors, actuators, generators, programmable logic controllers (PLCs), networking cables, and computers) [7]. More precisely, the cyber layer software is used to control different industrial processes in the physical layer through specific industrial networking protocols of Industrial IoT (IIoT). Plugging in the two layers to the Internet makes the entire SG network processes more vulnerable to cyberthreats. Cyberattacks are the attempts of sufficiently qualified individuals known as cybercriminals to destroy or maliciously use the cyberphysical system by targeting one of the CIA security pillars via unauthorised access to cybersecurity systems [8, 9]. Confidentiality attacks require unauthorised access via someone's credentials to private information with the purpose of malicious activity. Integrity attacks refer to intentional attacks that tend to modify data content leading to damaging the system [9, 10]. Availability attacks are time-delayed attacks that are usually Denial of Service (DoS) that tries to slow down data traffic and alter the whole process [11, 12]. Additionally, DoS attacks may also gain time to further proceed with confidentiality or integrity attacks [13].

Generally speaking, due to much vulnerability such as lack of authentication, data encryption, and continuous data integrity checking, SGs are vulnerable to one of the aforementioned threats under the

CIA umbrella. Therefore, necessary security, prevention, and process backup plans are top priorities for the cyberphysical system immunity against any possible adversary. In this context, cyberthreats detection and mitigation can be found under two categories; Human-Centric (HC) and Non-Human Centric (NHC) approaches [10]. HC approaches (i.e., authentication, training, passwords, awareness, and updates) refer to the involving of continuous training and expanding of human awareness about necessary new security precautions. NHC (i.e., blockchain, cloud computing, game theory [14], and ML, etc.) signify different modeling procedures and automatic detection via specifically designed hardware and software. Ongoing human awareness training and updates on safety precautions play an important role in preventing data theft. In fact, a simple daily mistake could be an easy cause of a data breach. Clicking wrong spam links, giving confidential information to inappropriate persons, or more generally, hesitantly ignoring security policies, are some common human errors, to mention a few [15].On the other hand, NHC approaches are very important in diagnosing (i.e. detecting and identifying) data traffic and making use of any suspicious false data symptoms when HC approaches are unable to cope with such digital threats. Blockchain digital ledgers are able to mitigate data changes, theft, or cheating by following certain specific rules for recording information. However, their installation and duplication across the entire SG nodes make it highly expensive especially in terms of energy consumption. Besides, private blockchains witnessed low-security efficiency precautions [16], [17], [18]. Cloud computing security-based has the advantage of allowing higher security features under low latency and computational costs. However, it needs higher bandwidth as well as it completely depends on the web service provider [19]. Although classical residue-based modeling and simulation techniques separate the two layers (i.e., cyber and physical layers), which are no longer effective in establishing the behavior of both simultaneously. Contrariwise, ML modeling procedures of attack behaviors based on historical data show promising performances and become leading alternatives in the cybersecurity field. It has the advantage of higher accuracy, ability to adapt to dynamic data, fewer deployment costs, and plenty of available blackbox models easy to be directly used [20].The motivations for industrial data processing have been the same for decades, namely to increase the revenue of investment by improving efficiency (i.e., by increasing productivity, and decreasing scrap, waste and energy usage), extending system lifetime, as well as enhancing safety and security. Sustainability has become yet another focal point of modern industry.

As people acknowledged the necessity of distributed data collection and massive data processing in various industrial areas, the research and innovation domain of IIoT (Industrial Internet of Things) began to thrive. Its business drive was promoted by the Industry 4.0 initiative, whereas its applications were extended from the very much overlapping Cyber-Physical Systems (CPS) domain. There is no generic, de-facto architecture for IIoT systems, although a layered approach is followed by domain experts. The purpose of splitting the layers could vary from communication types due to infrastructure need to the ecosystem stakeholder point of view; hence three, four or five layers can be identified provides a layered architectural view which shows the strong separation of technologies between the layers. It also indicates the different security approaches at the different layers [21].

While machine learning is exploited in various IIoT application areas, it is used only on a small subset of target areas extensively (see Figure 2). Depending on the application area, there are various purposes for processing industrial data. These include decision support, optimization, prediction, anomaly detection, classification, and clustering, just to name a few. In order to achieve the desired results, we need data—which are generally available for industrial players if IIoT-based data collection is in place—and we need physical resources for data processing—which are now available mostly due to the boom in GPU production. Because data and resources have been made available, we are able to use ML (Machine Learning) methods to achieve better results than ever before in the above areas. In terms of finding details on these methods, the first resources to turn to are, naturally,

textbooks. There are several great books on machine learning in general [22,23,24], and on modern tools regarding their application [25,26,27,28]. Further, we can find survey papers on utilizing machine learning in the industry. The authors of [29] provide a survey of the upcoming wave of machine learning in smart manufacturing. The specific topic of tackling faults by machine learning (ML) in the industry 4.0 era are surveyed in [30]. In a paper on machine learning multi-agent systems [31], the authors focus exclusively on their application in the oil and gas industry. Regarding different levels of industry 4.0, [32] focuses on ML methods applied in production planning and control. Similarly, a review of ML methods for the optimization of production processes is provided in [33].

To provide comparison with the topic of our current article, we can find more specific papers summarizing ML methods for smart production in general [34], or that review ML for production energy efficiency [35]. The authors of [36] provide a comprehensive overview of prognostic methods in the area of Industry 4.0. The authors of [37] focus on sustainability and predictive maintenance. Regarding reliability engineering and safety, the authors of [38] provide a targeted survey. Furthermore, ML support on safety assurance is surveyed in [39].

3. PROPOSED SYSTEM

The process begins by importing essential libraries, including Pandas for data manipulation, NumPy for numerical computations, Matplotlib and Seaborn for visualization, and scikit-learn modules for machine learning tasks. The dataset is then loaded from a CSV file, missing values are handled, and categorical variables are encoded using LabelEncoder. The data is split into features (X) and the target variable (y), followed by exploratory data analysis using count plots to visualize class distribution. Feature data is standardized using StandardScaler to ensure uniform scaling. Two machine learning models—LGBMClassifier and BaggingClassifier are trained on the processed training data. Their performance is evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrices for better insight into classification results.

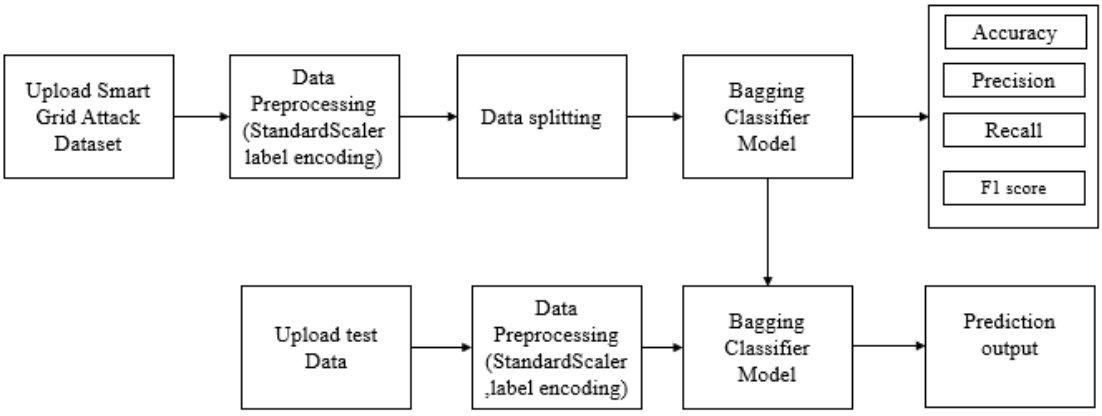

Fig. 2: Block Diagram.

Trained models are saved using joblib for future use. Model performance is compared to determine the best-performing algorithm. Finally, test data from "test.csv" is preprocessed similarly, and predictions are made using the trained models, with each test row printed alongside its predicted class (Attack or Normal).

Random Forest is a powerful supervised machine learning algorithm used for both classification and regression tasks. It leverages the ensemble learning technique of bagging

(bootstrap aggregating), where multiple decision trees are trained on random subsets of the dataset created through sampling with replacement. Each tree generates a prediction, and the final output is determined through majority voting (for classification) or averaging (for regression). This method improves accuracy, reduces overfitting, and handles large datasets efficiently. Key features include parallelization, resistance to the curse of dimensionality, robustness to missing data, and stability through aggregation. Random Forest introduces diversity by considering only a random subset of features at each split, making it more robust than standard Bagging Classifiers.

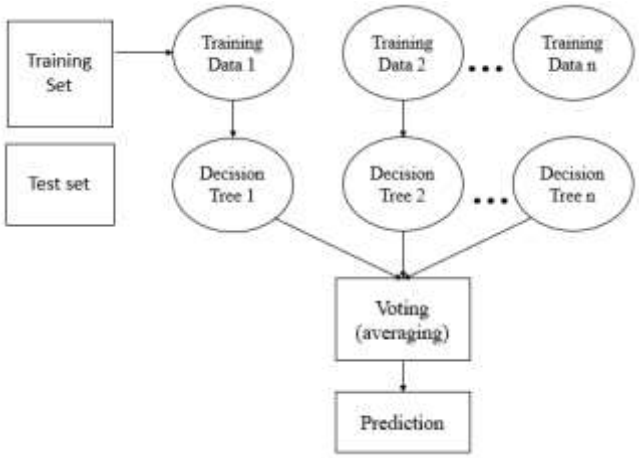

Fig. 3: Random Forest algorithm

While both Bagging and Random Forest reduce variance and improve generalization, Random Forest adds feature randomness to further decorrelate trees. Despite its computational cost, it performs well even in noisy environments. Random Forest is widely used in banking (loan risk), medicine (disease prediction), land use classification, and marketing (trend analysis), offering high accuracy, speed, and resilience in complex real-world applications.

4. RESULTS AND DISCUSSION

Figure 4 shows the

- **LGBMClassifier Accuracy:** This is the overall accuracy of the model on the test data. In this case, the accuracy is 75.27%, which means that the model correctly classified 75.27% of the instances in the test data.
- **LGBMClassifier Precision:** Precision is a metric that measures the proportion of positive predictions that were actually correct. A high precision means that most of the instances labeled as positive by the model were truly positive. In this case, the precision for the "Attack" class is 0.85 and 0.70 for the "Natural" class. This means that out of all the instances that the model predicted as "Attack", 85% were actually attacks, and out of all the instances predicted as "Natural", 70% were actually natural.
- **LGBMClassifier Recall:** Recall is a metric that measures the proportion of actual positive instances that were identified correctly by the model. A high recall means that the model was able to find most of the positive instances. In this case, the recall for the "Attack" class is 0.63 and 0.88 for the "Natural" class. This means that out of all the actual attack instances in the data, the model identified 63%, and out of all the actual natural instances, it identified 88%.

- **LGBMClassifier F1-score:** The F1-score is a harmonic mean of precision and recall. It's a way to balance between the two metrics and get a single measure of the model's performance. In this case, the F1-score for the "Attack" class is 0.72 and 0.78 for the "Natural" class.
- The weighted average at the bottom of the report shows the overall precision, recall, and F1-score across both classes, weighted by the number of instances in each class.
- The macro average is another way to calculate average precision, recall, and F1-score, but it gives equal weight to each class regardless of the number of instances.
- The support column shows the number of instances in each class. In this case, there are slightly more instances of the "Natural" class than the "Attack" class.

The classification report suggests that the LightGBM model is performing moderately well on this binary classification task. It has a good accuracy and is able to identify a reasonable proportion of both "Attack" and "Natural" instances. However, the recall for the "Attack" class is a bit lower than the recall for the "Natural" class, which suggests that the model might be missing some of the actual attack instances.

```
Model loaded successfully.
LGBMClassifier Accuracy    : 75.26929438818736
LGBMClassifier Precision   : 75.54490878633631
LGBMClassifier Recall      : 77.04654387056775
LGBMClassifier FSCORE      : 74.98182553794135

 LGBMClassifier classification report
              precision    recall  f1-score   support

      Attack       0.85      0.63      0.72     12669
     Natural       0.70      0.88      0.78     12118

    accuracy                           0.75     24787
   macro avg       0.77      0.76      0.75     24787
weighted avg       0.77      0.75      0.75     24787
```

Fig. 4: Classification report (LGBM Classifer)

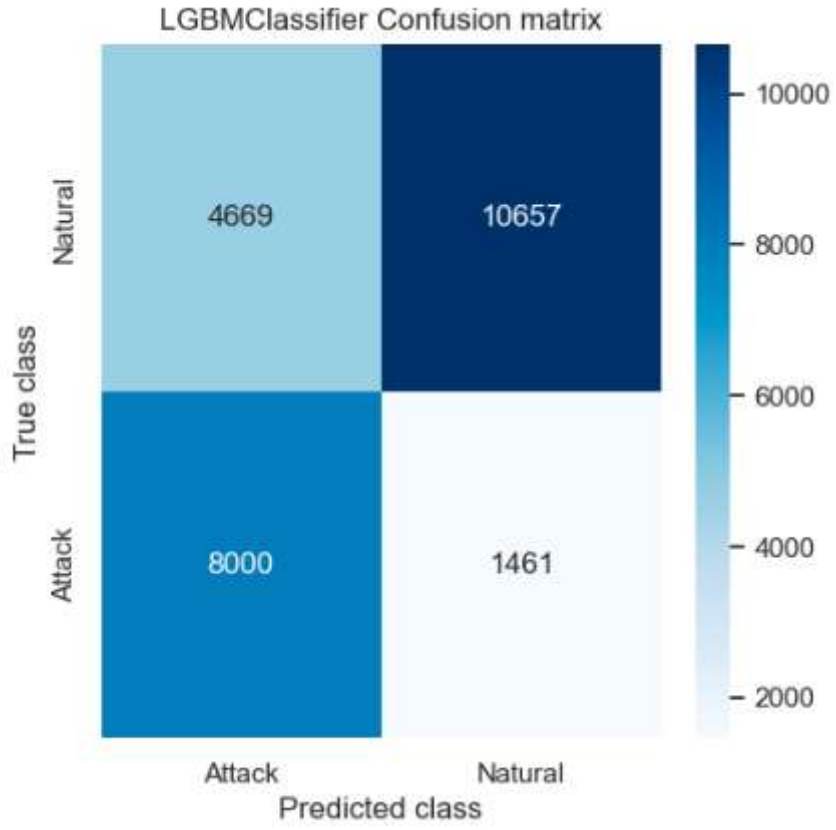

Fig. 5: Confusion Matrix **(**LGBM Classifier)

Figure 5 shows the

- The **Rows** represent the actual classes of the data samples. In this case, the rows represent "Natural" and "Attack".
- **Columns** represent the classes predicted by the model. In this case, the columns represent "Natural" and "Attack" classes as well.
- The values in each cell of the confusion matrix represent the number of data samples that belong to a particular class (row) but were predicted to belong to another class (column).
- **Top-left cell (4669):** This cell represents the number of data samples that were actually Natural and were correctly predicted as Natural by the model. There are 4669 of these samples.
- **Top-right cell (10657):** This cell represents the number of data samples that were actually Natural but were incorrectly predicted as Attack by the model. There are 10657 of these samples.
- **Bottom-left cell (8000):** This cell represents the number of data samples that were actually Attack but were incorrectly predicted as Natural by the model. There are 8000 of these samples.
- **Bottom-right cell (1461):** This cell represents the number of data samples that were actually Attack and were correctly predicted as Attack by the model. There are 1461 of these samples.

- The sum of the values in each row represents the total number of data samples in the corresponding actual class.
- The sum of the values in each column represents the total number of predictions made for the corresponding predicted class.
- Ideally, we want most of the values to be concentrated on the diagonal of the confusion matrix, where the actual class and the predicted class match. In this case, the model seems to be performing better at predicting Natural instances correctly (4669) compared to Attack instances (1461).
- This confusion matrix suggests that the model has a higher false positive rate for the "Attack" class (10657) than the "Natural" class (8000). This means that the model is misclassifying more Natural instances as Attack compared to Attack instances being misclassified as Natural.

```
Model loaded successfully.
BaggingClassifier Accuracy    : 94.00492193488522
BaggingClassifier Precision   : 93.96301694114226
BaggingClassifier Recall      : 94.08577532401159
BaggingClassifier FSCORE      : 93.99543980870794

 BaggingClassifier classification report
              precision    recall  f1-score   support

      Attack       0.93      0.96      0.94     12669
     Natural       0.95      0.92      0.94     12118

    accuracy                           0.94     24787
   macro avg       0.94      0.94      0.94     24787
weighted avg       0.94      0.94      0.94     24787
```

Fig. 6: Classification report (Bagging Classifier)

Figure 6 shows a classification report using a Bagging Classifier model in machine learning. It appears to be the result of a binary classification task, where the model is trying to distinguish between two classes: "Attack" and "Natural".

- **Model loaded successfully:** This message likely indicates that the Bagging Classifier model was successfully loaded from a file or created without errors.
- **BaggingClassifier Accuracy:** This is the overall accuracy of the model on the test data. In this case, the accuracy is 93.96%, which means that the model correctly classified 93.96% of the instances in the test data.
- **BaggingClassifier Precision:** Precision is a metric that measures the proportion of positive predictions that were actually correct. A high precision means that most of the instances labeled as positive by the model were truly positive. In this case, the precision for the "Attack" class is 0.93 and 0.95 for the "Natural" class. This means that out of all the instances that the model predicted as "Attack", 93% were actually attacks, and out of all the instances predicted as "Natural", 95% were actually natural.

- **BaggingClassifier Recall:** Recall is a metric that measures the proportion of actual positive instances that were identified correctly by the model. A high recall means that the model was able to find most of the positive instances. In this case, the recall for the "Attack" class is 0.96 and 0.92 for the "Natural" class. This means that out of all the actual attack instances in the data, the model identified 96%, and out of all the actual natural instances, it identified 92%.
- **BaggingClassifier F1-score:** The F1-score is a harmonic mean of precision and recall. It's a way to balance between the two metrics and get a single measure of the model's performance. In this case, the F1-score for the "Attack" class is 0.94 and 0.94 for the "Natural" class.

- The weighted average at the bottom of the report shows the overall precision, recall, and F1-score across both classes, weighted by the number of instances in each class.
- The macro average is another way to calculate average precision, recall, and F1-score, but it gives equal weight to each class regardless of the number of instances.
- The support column shows the number of instances in each class. In this case, there are almost the same number of instances of the "Natural" class (12118) and the "Attack" class (12669).

The classification report suggests that the Bagging Classifier model is performing well on this binary classification task. It has a high accuracy and is able to identify a significant proportion of both "Attack" and "Natural" instances. The precision, recall, and F1-score are all very similar across the two classes, indicating a balanced performance.

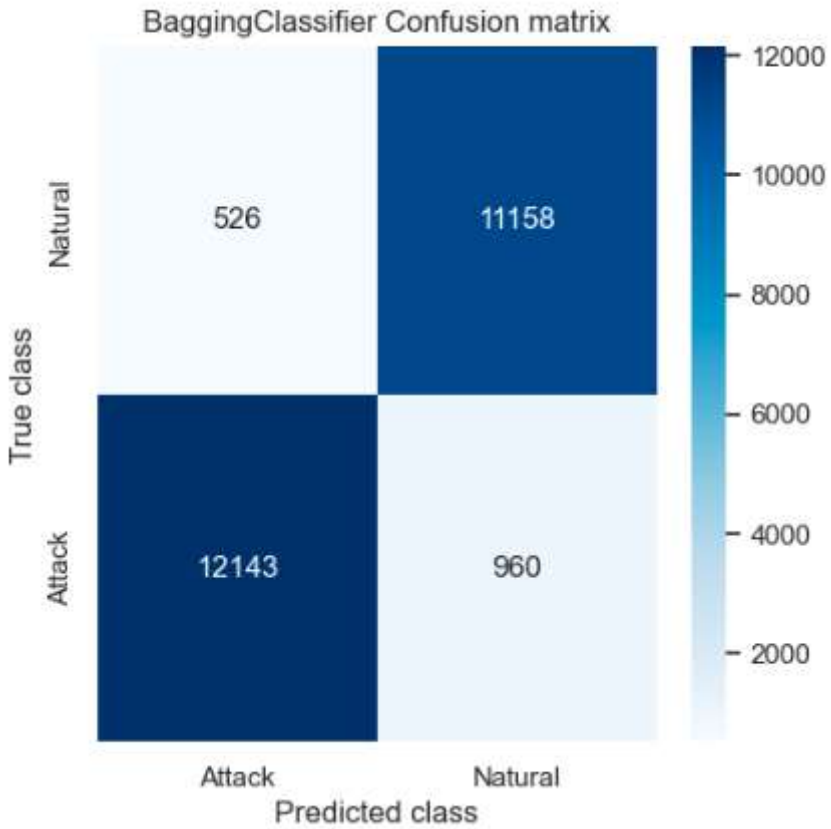

Fig. 7: Confusion Matrix (Bagging Classifier)

Figure 7 shows the

- **Top-left cell (12000):** This cell represents the number of data samples that were actually Natural and were correctly predicted as Natural by the model. There are 12000 of these samples.
- **Top-right cell (1000):** This cell represents the number of data samples that were actually Natural but were incorrectly predicted as Attack by the model. There are 1000 of these samples.
- **Bottom-left cell (4000):** This cell represents the number of data samples that were actually Attack but were incorrectly predicted as Natural by the model. There are 4000 of these samples.
- **Bottom-right cell (526):** This cell represents the number of data samples that were actually Attack and were correctly predicted as Attack by the model. There are 526 of these samples.
- The sum of the values in each row represents the total number of data samples in the corresponding actual class.
- The sum of the values in each column represents the total number of predictions made for the corresponding predicted class.
- Ideally, we want most of the values to be concentrated on the diagonal of the confusion matrix, where the actual class and the predicted class match. In this case, the model seems to be performing well at predicting both Natural (12000) and Attack (526) instances correctly.
- This confusion matrix suggests that the model has a lower false positive rate (1000) for the "Natural" class compared to the "Attack" class (4000). This means that the model is misclassifying more Attack instances as Natural compared to Natural instances being misclassified as Attack.

	Algorithm Name	Precison	Recall	FScore	Accuracy
0	LGBMClassifier	75.544909	77.046544	74.981826	75.269294
1	BaggingClassifier	89.124371	89.844615	89.188589	89.256465

Fig. 8: Comparison of LGBM & Bagging Classifier

Figure 8 shows the Bagging Classifier outperforms the LightGBM Classifier across multiple key metrics:

Accuracy: Bagging Classifier achieves an accuracy of 89.26%, significantly higher than the LightGBM Classifier's 75.27%. This implies that the Bagging Classifier makes nearly 14% more correct predictions than the LightGBM Classifier.

Precision: For both "Attack" and "Natural" classes, the Bagging Classifier demonstrates higher precision compared to the LightGBM Classifier. Specifically, it achieves a precision of 0.93 for "Attack" and 0.95 for "Natural," while the LightGBM Classifier achieves 0.85 and 0.70 for "Attack" and "Natural," respectively.

Recall: The Bagging Classifier exhibits higher recall for the "Attack" class (0.96) compared to the LightGBM Classifier (0.63). However, it slightly lags behind in recall for the "Natural" class (0.92) compared to the LightGBM Classifier's 0.88.

F1-Score: Across both classes, the Bagging Classifier attains substantially higher F1-scores compared to the LightGBM Classifier. For the "Attack" class, it achieves 0.94, while for the "Natural" class, it also reaches 0.94. In contrast, the LightGBM Classifier achieves F1-scores of 0.72 and 0.78 for "Attack" and "Natural," respectively.

The Bagging Classifier demonstrates superior performance across all metrics, making it the preferable choice for this classification task. However, it's essential to consider that the choice of the best model can depend on the specific dataset and task requirements.

5. CONCLUSION

The integration of machine learning into smart grid maintenance significantly enhances the proactive management of Intelligent Electronic Device (IED) failures. This study implemented a structured framework involving data preprocessing, including handling missing values and encoding categorical variables, followed by exploratory data analysis to understand failure patterns. Classification models such as RandomForestClassifier, BaggingClassifier, and LGBMClassifier were trained on a balanced dataset using SMOTE to address class imbalance. Evaluation using accuracy, precision, recall, and F1-score revealed that LGBMClassifier and BaggingClassifier delivered strong performance, effectively distinguishing between failure types as shown by their confusion matrices. The trained models were saved for deployment, enabling real-time failure prediction and facilitating timely maintenance. With its scalability and high accuracy, the framework is well-suited for integration into smart grid systems, offering utility companies a reliable tool for improving operational efficiency and reducing downtime.

REFERENCES

[1] Machine learning applications in cascading failure analysis in power systems: Sami, N.M., Naeini, M. 2024 Electric Power Systems Research

[2] A review on machine learning techniques for secured cyber-physical systems in smart grid networks Hasan, M.K., Abdulkadir, R.A., Islam, S., Gadekallu, T.R., Safie, N. 2024 Energy Reports

[3] Multi-aspect rule-based AI: Methods, taxonomy, challenges and directions towards automation, intelligence and transparent cybersecurity modeling for critical infrastructures

Sarker, I.H., Janicke, H., Ferrag, M.A., Abuadbba, A. 2024 Internet of Things (Netherlands)

[4] Reinforcement Learning-Empowered Graph Convolutional Network Framework for Data Integrity Attack Detection in Cyber-Physical Systems

Vincent, E., Korki, M., Seyedmahmoudian, M., Stojcevski, A., Mekhilef, S. 2024 CSEE Journal of Power and Energy Systems

[5] Smart Grid Security: An Effective Hybrid CNN-Based Approach for Detecting Energy Theft Using Consumption Patterns

Gunduz, M.Z., Das, R. 2024 Sensors

[6] Intelligent learning approaches for demand-side controller for BIPV-integrated buildings (Book Chapter)

Liu, Z., Zhang, L., Wang, S. 2024 Intelligent Learning Approaches for Renewable and Sustainable Energy

[7] Optimization of Cloud Migration Parameters Using Novel Linear Programming Technique

Afzal, S., Thakur, A., Singh, P. 2024 Lecture Notes in Electrical Engineering

[8] A Review on the Evaluation of Feature Selection Using Machine Learning for Cyber-Attack Detection in Smart Grid

Mohammed, S.H., Al-Jumaily, A., Singh, M.S.J., (...), Al-Najjar, M.M.A.K., Al-Jumeily, D.2024 IEEE Access

[9] Deep Learning Based Bug Detection in Solidity Smart Contracts

Singh, J., Sahu, D.P., Murkute, S., (...), Agarwal, M., Kumar, P. 2024 Communications in Computer and Information Science

[10] Comparative Analysis of Various Transfer Learning Approaches in Deep CNNs for Image Classification Tyagi, A., Khandelwal, R., Shelke, N.A., (...), Rajpal, D., Gaware, I.R.2024 communications in Computer and Information Science

[11] Various Active Learning Strategies Analysis in Image Labeling: Maximizing Performance with Minimum Labeled Data

Tyagi, A., Aditya, H., Shelke, N.A., (...), Jadeja, Y., Turukmane, A.V. 2024 Communications in Computer and Information Science

[12] Securing Smart Grids Using Machine Learning Algorithms (Book Chapter)

Das, M., Sood, V.M., Garg, K.D., Narang, S.K. 2024 Artificial Intelligence and Society 5.0: Issues, Opportunities, and Challenges

[13] A comprehensive review of AI-enhanced smart grid integration for hydrogen energy: Advances, challenges, and future prospects

SaberiKamarposhti, M., Kamyab, H., Krishnan, S., (...), Chelliapan, S., Khorami, M. 2024 International Journal of Hydrogen Energy

Articles not published yet, but available online Article

[14] Soft computing based smart grid fault detection using computerised data analysis with fuzzy machine learning model

Chen, T., Liu, C. 2024 Sustainable Computing: Informatics and Systems

[15] The detection and prevention of phishing threats in OSN using machine learning techniques (Book Chapter)

Samal, S., Mohanty, S., Acharya, A.A. 2023 Building Intelligent Systems Using Machine Learning and Deep Learning: Security, Applications and Its Challenges

[16] Detecting and classifying man-in-the-middle attacks in the private area network of smart grids

Elrawy, M.F., Hadjidemetriou, L., Laoudias, C., Michael, M.K. 2023 Sustainable Energy, Grids and Networks

[17] Multiverse Recurrent Expansion With Multiple Repeats: A Representation Learning Algorithm for Electricity Theft Detection in Smart Grids

Berghout, T., Benbouzid, M., Ferrag, M.A. 2023 IEEE Transactions on Smart Grid

[18] Data-driven prediction models of photovoltaic energy for smart grid applications

Open Access

Souabi, S., Chakir, A., Tabaa, M. 2023 Energy Reports

[19] Towards Resilient and Secure Smart Grids against PMU Adversarial Attacks: A Deep Learning-Based Robust Data Engineering Approach

Berghout, T., Benbouzid, M., Amirat, Y. 2023 Electronics (Switzerland)

[20] An Adaptive Deep Learning Neural Network Model to Enhance Machine-Learning-Based Classifiers for Intrusion Detection in Smart Grids

Sami, N.M., Naeini, M. 2024 Electric Power Systems Research

Certainly, here's the list with the numbering in squared brackets:

[21]. Varga P., Plosz S., Soos G., Hegedus C. Security threats and issues in automation IoT; Proceedings of the 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS); Trondheim, Norway. 31 May–2 June 2017; pp. 1–6. [CrossRef] [Google Scholar]

[22]. Mitchell T.M. Machine Learning. McGraw-Hill Education; New York, NY, USA: 1997. [Google Scholar]

[23]. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press; Cambridge, MA, USA: 2016. [(accessed on 1 November 2022)]. Available online: http://www.deeplearningbook.org [Google Scholar]

[24]. Shalev-Shwartz S., Ben-David S. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press; Cambridge, UK: 2014. [Google Scholar]

[25]. Géron A. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc.; Sebastopol, CA, USA: 2022. [Google Scholar]

[26]. Müller A.C., Guido S. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, Inc.; Sebastopol, CA, USA: 2016. [Google Scholar]

[27]. Lantz B. Machine Learning with R: Expert Techniques for Predictive Modeling. Packt Publishing Ltd.; Birmingham, UK: 2019. [Google Scholar]

[28]. Lakshmanan V., Robinson S., Munn M. Machine Learning Design Patterns. O'Reilly Media, Inc.; Sebastopol, CA, USA: 2020. [Google Scholar]

[29]. Sharp M., Ak R., Hedberg T., Jr. A survey of the advancing use and development of machine learning in smart manufacturing. J. Manuf. Syst. 2018;48:170–179. doi: 10.1016/j.jmsy.2018.02.004. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[30]. Angelopoulos A., Michailidis E.T., Nomikos N., Trakadas P., Hatziefremidis A., Voliotis S., Zahariadis T. Tackling faults in the industry 4.0 era—A survey of machine-learning solutions and key

aspects. Sensors. 2019;20:109. doi: 10.3390/s20010109. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[31]. Hanga K.M., Kovalchuk Y. Machine learning and multi-agent systems in oil and gas industry applications: A survey. Comput. Sci. Rev. 2019;34:100191. doi: 10.1016/j.cosrev.2019.08.002. [CrossRef] [Google Scholar]

[32]. Usuga Cadavid J.P., Lamouri S., Grabot B., Pellerin R., Fortin A. Machine learning applied in production planning and control: A state-of-the-art in the era of industry 4.0. J. Intell. Manuf. 2020;31:1531–1558. doi: 10.1007/s10845-019-01531-7. [CrossRef] [Google Scholar]

[33]. Weichert D., Link P., Stoll A., Rüping S., Ihlenfeldt S., Wrobel S. A review of machine learning for the optimization of production processes. Int. J. Adv. Manuf. Technol. 2019;104:1889–1902. doi: 10.1007/s00170-019-03988-5. [CrossRef] [Google Scholar]

[34]. Cioffi R., Travaglioni M., Piscitelli G., Petrillo A., De Felice F. Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. Sustainability. 2020;12:492. doi: 10.3390/su12020492. [CrossRef] [Google Scholar]

[35]. Narciso D.A., Martins F. Application of machine learning tools for energy efficiency in industry: A review. Energy Rep. 2020;6:1181–1199. doi: 10.1016/j.egyr.2020.04.035. [CrossRef] [Google Scholar]

[36]. Diez-Olivan A., Del Ser J., Galar D., Sierra B. Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. Inf. Fusion. 2019;50:92–111. doi: 10.1016/j.inffus.2018.10.005. [CrossRef] [Google Scholar]

[37]. Çınar Z.M., Abdussalam Nuhu A., Zeeshan Q., Korhan O., Asmael M., Safaei B. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustainability. 2020;12:8211. doi: 10.3390/su12198211. [CrossRef] [Google Scholar]

[38]. Xu Z., Saleh J.H. Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. Reliab. Eng. Syst. Saf. 2021;211:107530. doi: 10.1016/j.ress.2021.107530. [CrossRef] [Google Scholar]

[39]. Schwalbe G., Schels M. A survey on methods for the safety assurance of machine learning based systems; Proceedings of the 10th European Congress on Embedded Real Time Software and Systems (ERTS 2020); Toulouse, France. 29–31 January 2020. [Google Scholar]

[40]. Martin R., Schrecker S., Soroush H., Molina J., LeBlanc J., Hirsch F., Buchheit M., Ginter A., Banavara H., Eswarahally S., et al. Industrial Internet Security Framework Technical Report. CreateSpace Independent Publishing Platform; Scotts Valley, CA, USA: 2016. Technical Report. [CrossRef] [Google Scholar]