



Stacking Ensemble Learning Model for Intrusion Detection in Electrical Substation

Mohammad Mahruf Alam¹, Feddy Setio Pribadi², Rizky Ajie Aprilianto³, Arvina Rizqi Nurul'aini⁴
^{1,2,3,4}Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Semarang, Semarang, Indonesia

¹mahrufalam99@students.unnes.ac.id, ²feddy.setio@mail.unnes.ac.id, ³rizkyajiea@mail.unnes.ac.id,

⁴arvinarizqi@students.unnes.ac.id

Abstract

Electrical substations are crucial infrastructure in power transmission and distribution but are increasingly vulnerable to cyber threats. However, existing intrusion detection systems (IDS) face several limitations, such as high false positive rates, weak in anticipating new attack patterns, and imbalances in detecting different types of intrusions. This study proposes a Stacking Ensemble Learning model to enhance intrusion detection accuracy in electrical substations. The proposed model integrates Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and XGBoost (XGB) as base models with XGB acting as the meta-model. A real-world electrical substation IEC 60870-5-104 network traffic dataset comprising 319,949 instances with multiple attacks, such as DoS, Port Scan, NTP DDoS, IEC 104 Starvation, Fuzzy Attack, Flood Attack, and MITM, was used in this study. The results demonstrate that the stacking model achieves the best performance, with accuracy (0.99990), precision (0.99990), recall (0.99990), and F1 score (0.99990), surpassing the base model, Bagging, and Boosting. T-test results further confirmed statistical significance, with p-values of 0.00428 (LR), 0.04237 (SVM), 0.00000 (XGB), 0.00057 (KNN), 0.00549 (Boosting), and 0.00000 (Bagging) reinforcing the superiority of the proposed method approach. These findings highlight the effectiveness of Stacking Ensemble Learning in enhancing the detection performance of IDS for electrical substations and outperforming traditional models and other ensemble learning methods.

Keywords: electrical substations; intrusion detection system; machine learning; stacking ensemble learning

How to Cite: M. M. Alam and F. S. Pribadi, "Stacking Ensemble Learning Model for Intrusion Detection in Electrical Substation", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 5, pp. 1030 - 1037, Oct. 2025.

Permalink/DOI: <https://doi.org/10.29207/resti.v9i5.6502>

Received: March 21, 2025

Accepted: July 23, 2025

Available Online: October 11, 2025

This is an open-access article under the CC BY 4.0 License

Published by Ikatan Ahli Informatika Indonesia

1. Introduction

Electrical substations have become an essential part of power transmission and distribution systems. They are important to ensure stability and reliability in the power system [1]. Modern substations are now interconnected through the networks, enabling seamless communication, automation, and remote monitoring [2]. The most common protocols used in substation networks are IEC 61850 and IEC 60870-104. IEC 61850 creates fast and standardized communication between Intelligent Electronic Devices (IEDs) [3], while IEC 60870-104 facilitates remote monitoring and control over wide-area networks [4].

However, the growth of this digitalization poses a challenge as substations become more vulnerable to network intrusion and cyber-attacks. This attack can cause serious damage to the substation system as hackers could take control of the substation through the

network [5]. Hence, an intrusion detection systems (IDS) are essential for the security, reliability, and resilience of a critical infrastructure such as substations [6].

The current IDS system is divided into three categories: signature-based IDS, anomaly-based IDS, and hybrid IDS [7]. Signature-based IDS are developed by giving predefined attack signatures for the system to identify malicious data by matching it to known patterns [8]. Anomaly-based IDS are developed by training models to learn normal behavior and give alerts when anomalies arise [9]. Then, hybrid IDS combines both methods for more effective intrusion detection. Although IDS has become more advanced, challenges and limitations exist [10], such as low accuracy in detecting intrusion and high false positive rates [11].

Several studies have been conducted on IDS development. A study by Zhu et al. utilized the Random

Forest (RF) for detecting intrusions in electrical power systems and achieved 99.39% average accuracy and 0.037% false alarm rate [12]. Other studies by Shanti et al. utilized the Support Vector Machine (SVM) and Isolation Forest Model (IFM) for IDS and achieved 0.99 accuracies for the Isolation Forest and 0.95 accuracies for the SVM [13]. Their study also reported additional evaluation metrics, with the Isolation Forest model obtaining recall score of 0.87 and F1-score of 0.78, while the SVM achieved a recall score of 0.88 and F1-score of 0.67. However, their study did not include false alarm results. In addition, a study by Sharma and Shah utilized a hybrid approach of RF and SVM, achieving an accuracy of 98.98%, precision of 97.9%, recall score of 97.6%, and F1-score of 99.9% [14].

From previous studies, existing models still face challenges such as high false positives, limited adaptability to detect zero-day attack, and imbalanced detection rates across different intrusion types. This problem highlights the need for improved detection performance, false alarm reduction, and model generalization.

Machine Learning (ML) approach has become a key method for developing IDS [15]. ML can extract meaningful patterns from data, allowing it to make informed judgments and accurate predictions. This approach is essential for detecting network intrusion and enhancing decision-making. There are two popular ML approaches: supervised and unsupervised learning [16]. Supervised learning is ML algorithms that learn patterns from input data and map the target using labeled output data to make predictions [17]. Unsupervised learning is an ML algorithm that discovers patterns in data without requiring labeled outputs [18]. Both approaches are powerful methods in the development of IDS.

Traditional machine learning models, including individual classifiers and hybrid approaches, often rely on a single or limited combination of algorithms, which may not be fully effective in learning the complexity of network intrusions and preventing it [19]. Stacking Ensemble Learning offers a promising approach to address these limitations by combining multiple ML model with a meta-model to enhance performance and robustness [20]. Unlike traditional ensemble methods such as Bagging and Boosting, stacking utilizes the diversity of different ML algorithms by training them individually and then using a high-level model to learn from their combined predictions [21]. This hierarchical learning process allows the model to mitigate individual weaknesses, improve generalization, and capture complex patterns more effectively, leading to a more adaptive and reliable IDS. Given network traffic's diverse and evolving nature, IDS data often exhibits complex, nonlinear patterns that require sophisticated learning techniques to distinguish malicious activities accurately.

Based on the explanation, this study proposes a highly accurate and robust IDS using Stacking Ensemble Learning, integrating Logistic Regression (LR), SVM, K-Nearest Neighbors (KNN), and XGBoost (XGB). The study focuses on enhancing performance and reducing false alarm using a real-world IEC 60870-104 dataset, enabling a realistic evaluation of the model under practical conditions. By leveraging the strengths of LR, SVM, and KNN in capturing different decision boundaries and data distributions, combined with XGB's ability to handle complex patterns, the proposed model aims to provide a more effective and robust IDS that can be applicable in electrical substations while outperforming the performance and robustness of standard ensemble methods such as Bagging and Boosting.

2. Methods

Figure 1 visualizes the research method of this study. This study was conducted in four stages: data collecting, data preprocessing, model training, and model evaluation.

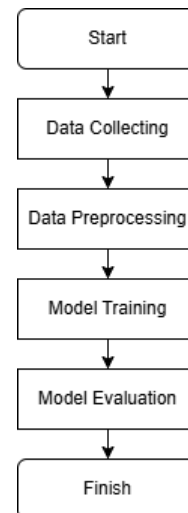


Figure 1. Research Method

Figure 2 illustrates the system design, which outlines the end-to-end pipeline from data preprocessing to model evaluation. The explanation of the system design is shown in section 2.1 to 2.6.

2.1 Dataset

The dataset used in this study was obtained from the Data in Brief journal, specifically from the study by Gutiérrez et al. [22], titled “A dataset to train intrusion detection systems based on machine learning models for electrical substations,” published in December 2024. The dataset used in the study can be found at <https://zenodo.org/records/14096593>. This study focuses on the dataset of IEC 60870-104 (IEC104), which is a protocol widely used in electrical substations. The dataset includes both real substation traces and testbed traces. The real substation traces were collected

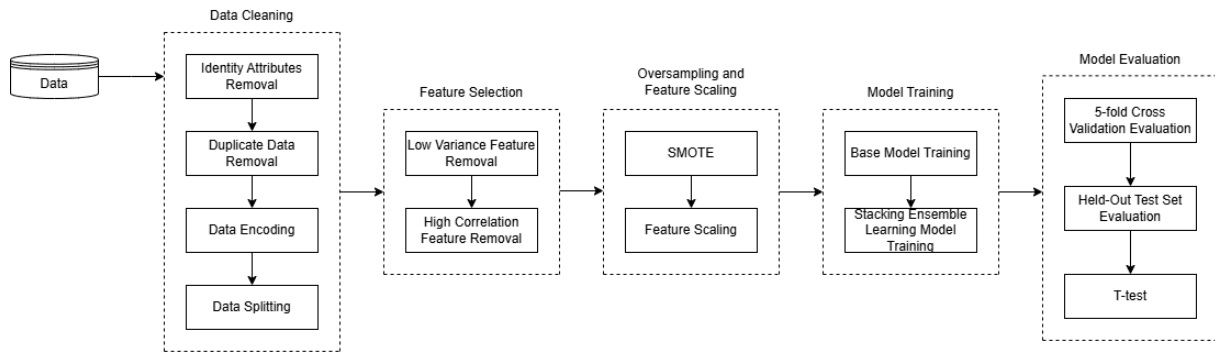


Figure 2. System Design

from an electrical facility in Iltsi, Ukraine, where IEC104 and NTP communications network traffic was captured over seven days. A controlled laboratory testbed was implemented to simulate the attack scenarios, as performing such tests in real substations is infeasible due to the critical nature of the infrastructure.

A total of 83 features with 319,949 records consist in the dataset covering various attack scenarios, including Denial of Service (DoS), Flood attack, Fuzzy attack, IEC 104 starvation attack, Man-in-the-Middle (MITM), NTP DDoS, Port scan attacks, and normal condition-making it highly relevant for cybersecurity research. These features include numerical and categorical attributes, such as network identifiers, traffic flow statistics, time-based features, and protocol-related attributes. Additionally, throughput, performance metrics, and bulk transfer provide deeper insights into network traffic patterns. These attributes are crucial in distinguishing between normal and malicious traffic, enabling more accurate intrusion detection.

One of the key challenges of this dataset is its imbalanced class distribution, where certain attack types, such as MITM and Flood Attack, have significantly fewer instances than others, like DoS. This imbalance can negatively impact model performance, making it crucial to apply techniques such as resampling to enhance classification accuracy. Furthermore, the dataset reflects real-world cybersecurity threats, making it well-suited for testing and improving intrusion detection models. The class distribution for attack scenarios is shown in Table 1.

Table 1. Attack Scenarios Data Distribution

Scenarios	Total
DoS	304627
Port Scan	9710
NTP DDoS	2278
IEC 104 Starvation	2028
Fuzzy Attack	939
Attack Free	255
Flood Attack	108
MITM	26

2.2 Data Cleaning

The dataset underwent several preprocessing steps to enhance model performance [23]. Irrelevant identity-based attributes were removed from the dataset To

improve the efficiency and accuracy of the model. These attributes, such as Flow ID, Src IP, Dst IP, Dst Port, Protocol, and Timestamp, do not contribute meaningful information for intrusion detection and could introduce unnecessary complexity. Additionally, duplicate records were removed to prevent data redundancy, ensuring that the model is trained on diverse network traffic patterns and reducing the risk of overfitting.

Then, different attack types were converted into numerical values. This transformation ensures compatibility with machine learning algorithms and effectively allows the model to learn from attack classifications. Encoding the labels also facilitates accurate predictions by assigning meaningful numerical representations to each category.

After that, the dataset was divided into 70:30 ratio for train and test set. This ensure a fair model performance evaluation. Stratified sampling was applied to maintain the original class distribution, preventing any training and testing datasets imbalance.

2.3. Feature Selection

Feature selection is an important step in model development that enhances performance by eliminating irrelevant or redundant features [24]. This study used a filter-based method for feature selection as it efficiently removes irrelevant features before model training, reducing computational costs and improving performance [25].

Feature selection begins with removing constant features, which have zero variance and provide no valuable information for distinguishing between data points. Next, quasi-constant features, which exhibit minimal variance with a 0.01 threshold, are removed to prevent redundancy. Highly correlated features are identified and removed using Spearman's correlation matrix to further refine the dataset. Feature from each highly correlated pair with a correlation coefficient above 0.9 is dropped to reduce multicollinearity.

2.4 Oversampling and Feature Scaling

After feature selection, SMOTE techniques were applied to tackle the class imbalance problem in the dataset. This technique increases the representation of

minority classes by creating synthetic samples, ensuring a more balanced class distribution [26]. By doing so, the model avoids bias toward majority classes and improves its ability to detect underrepresented attack types.

Lastly, MinMax Scaling was applied to standardize all numerical features. This scaling technique transforms feature values to the 0 and 1 range, preserving the original distribution while ensuring no feature dominates due to its scale [27]. MinMax Scaling improves model convergence by preventing large-magnitude features from skewing the learning process. This normalization step enhances training stability and ensures that all features contribute proportionally to intrusion detection. Formula 1 shows the formula for calculating MinMax scaling.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

x is the data value.

2.5 Model Training

The proposed model utilizes Stacking Ensemble Learning, which integrates LR, SVM, XGB, and KNN as base models, with XGB serving as the meta-model for more reliable and accurate intrusion detection. Each base model is trained separately before being combined in the stacking framework, allowing for a comprehensive comparison between individual models and the ensemble approach.

Model Training is conducted on Google Colab using Python 3.11.11. Several essential libraries used in this study are detailed in Table 2.

Table 2. Libraries Used

Function	Library
Data Manipulation	Pandas, NumPy
Visualization	Matplotlib, Seaborn
Model Training and Evaluation	Scikit-learn

Stacking Ensemble Learning is an advanced ensemble learning method that enhances ML performance by combining multiple ML models as the base models with a meta-model [28]. This method enhances the meta-model's generalization and improves prediction accuracy and robustness [29]. In this study, base models such as LR, SVM, XGB, and KNN are trained on the dataset to generate individual predictions. Out-of-fold (OOF) predictions from each base model are generated using 5-fold cross-validation to avoid data leakage. These OOF predictions are then used as input to the meta-model. XGB performs the final classification as the meta-model. Figure 3 illustrates the proposed Stacking Ensemble Learning model architecture.

LR is used in this study as a baseline model because of its simplicity, efficiency, and ability to provide probabilistic outputs [30]. IDS requires models that can quickly classify anomalies in network traffic, making LR a practical starting point. It performs well with linearly separable data, which helps identify straightforward attack patterns.

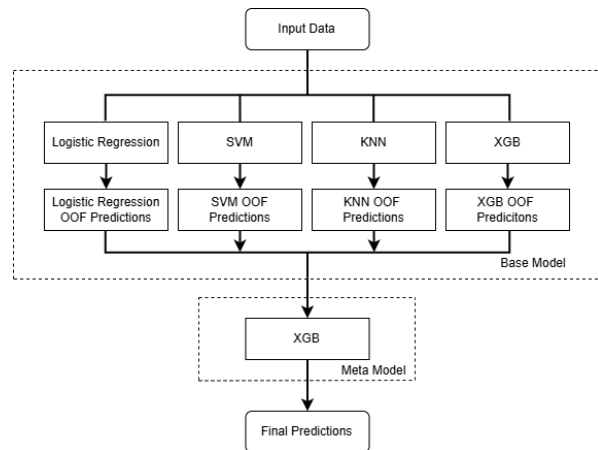


Figure 3. Proposed Stacking Ensemble Learning Model Architecture

SVM is included in this study due to its effectiveness in handling complex high-dimensional data. The dataset used in this research contains 83 features, many of which may exhibit complex relationships that define attack patterns. SVM is known for finding optimal decision boundaries by maximizing the margin between different classes [31], making it highly effective in distinguishing network traffic. Furthermore, the flexibility of using kernel functions enables SVM to capture nonlinear relationships within the data [32].

KNN is selected for its instance-based learning approach [33], which proves highly effective for intrusion detection. Unlike parametric models, KNN does not require strict assumptions about data distribution, making it more flexible in recognizing various cyberattack patterns. Given our dataset, which consists of 319,949 entries and 83 features, KNN can leverage its non-parametric nature to model complex attack patterns. KNN classifies network traffic by comparing it against previously observed instances, thereby making it particularly adept at detecting novel threats.

XGB is utilized as the meta-model in the stacking ensemble framework because it combines predictions from multiple base learners and improves classification accuracy. XGB is highly efficient due to its optimized implementation, which includes parallel processing, tree pruning, and hardware-aware computation [34]. XGB's gradient Boosting framework also helps reduce bias and variance, leading to a more generalized model. As the meta-model, XGB optimally combines the outputs of the base models to produce more reliable intrusion detection results based on their strength.

2.6 Model Evaluation

Model evaluation was conducted using several commonly used performance metrics such as accuracy, precision, recall, F1-score, and t-test to assess the performance of the Stacking Ensemble Model. Accuracy measures the correct prediction among the dataset. Precision calculated the number of correctly identified intrusion cases among predicted intrusions [35]. Recall assesses the model's performance to detect

all actual intrusion cases. The F1-score combines precision with recall and then divide them, providing a stable metrics between the two. This makes it a handy metric, particularly when applied to datasets with class imbalance. The performance metrics are calculated using Formula 2,3, and 4.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

where TP are intrusions cases that correctly predicted, FP are cases incorrectly predicted as intrusions, and FN represents intrusions cases that predicted as normal.

These performance metrics ensure a comprehensive analysis of the model's performance in intrusion detection. In addition to the performance metrics, the model evaluation also includes a confusion matrix to provide a detailed view of the model's classification performance.

The t-test is a statistical method used to identify whether there is a significant difference between two independent groups. In this study, the t-test was applied to compare the performance metrics of the stacking model with the base models, Bagging and Boosting. The null hypothesis (H_0) assumes no significant difference exists between the Stacking Ensemble Learning Model with the base, Bagging, and Boosting models, whereas the alternative hypothesis (H_1) suggests a significant difference [36]. The t-test statistic can be compute computed using Formula 5.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (5)$$

$\bar{X}_1$ and $\bar{X}_2$ represent the performance metrics of the two models, s_1^2 and s_2^2 are the variances of performance metrics for each model, and n_1 and n_2 are the number of metrics. A significance level of $\alpha = 0.05$ was used to determine statistical significance. If the resulting p-value is less than 0.05, H_0 is rejected, indicating a statistically significant difference exist. This analysis provides insights into whether the performance improvements of the Stacking Model are statistically meaningful.

Additionally, a 5-fold cross-validation (CV) strategy was employed to prevent overfitting while ensuring generalization capabilities. This approach had two main objectives: (1) to generate OOF predictions from the base model, which were then used as train data for the meta-model, and (2) to provide consistent performance evaluations for all models. 5-fold CV was applied to each model to obtain average metrics from various data divisions. After the training and cross-validation processes were completed, the final performance was evaluated using a separate test set to assess the model's effectiveness on new data that had not been used before.

3. Results and Discussions

The proposed Stacking Ensemble Learning, base, Bagging, and Boosting model was evaluated using performance and t-test. Each model was assessed using 5-fold CV on the training set to ensure consistent performance across different data splits. During the 5-fold CV process, each fold contained approximately 341,163 samples, ensuring that every model was trained and validated on balanced and representative subsets of the data. The average cross-validation metrics are reported in Table 3.

Table 3. Model Evaluation Using 5-Fold CV

Model	Accuracy	Precision	Recall	F1-Score
Boosting	0.85701	0.89384	0.85701	0.83760
LR	0.95408	0.96249	0.95408	0.95292
SVM	0.95265	0.96196	0.95265	0.95060
XGB	0.99991	0.99991	0.99991	0.99991
KNN	0.99992	0.99992	0.99992	0.99992
Bagging	0.99992	0.99992	0.99992	0.99992
Stacking				
Ensemble Learning	0.99994	0.99994	0.99994	0.99994

The 5-fold CV results show that Stacking Ensemble Learning delivers the best performance on all evaluation metrics, with performance metrics score of 0.99994, surpassing all other models. Bagging and KNN followed, yielding nearly equivalent results, with each achieving 0.99992 on all performance metrics, indicating highly reliable and consistent classification capabilities. XGB also performed well with a score of 0.99991 on all metrics. On the other hand, SVM achieved an accuracy and recall of 0.95265, a precision of 0.96196, and an F1-score of 0.95060. LR was slightly lower but still competitive, with an accuracy and recall of 0.95408, precision of 0.96249, and an F1-score of 0.95292. Boosting ranked lowest among the models tested, with an accuracy and recall of 0.85701, precision of 0.89384, and an F1-score of 0.83760.

After CV, the final models were later tested on the held-out test set to assess their generalization ability. These results are shown in Table 4.

Testing results on a held-out test set show that Stacking Ensemble Learning once again ranks at the top across all performance metrics, with an accuracy, recall and F1-score of 0.99989, and precision of 0.99990, surpassing all other models. KNN came in second with a very close performance, namely an accuracy and recall of 0.99984, a precision of 0.99986, and an F1-score of 0.99985, confirming its classification strength. Bagging also showed excellent results with an accuracy, recall, and F1-score of 0.99980, and precision of 0.99981. Meanwhile, XGB recorded comparable performance, with an accuracy and recall of 0.99954, precision of 0.99956, and an F1-score of 0.99955. Boosting achieved an accuracy and recall of 0.99786, precision of 0.99884, and an F1-score of 0.99833. SVM performed quite competitively with an accuracy and recall of 0.98843, precision of 0.99848, and an F1-score of 0.99201. LR showed the lowest performance among

all models, with an accuracy and recall of 0.97718, precision of 0.98730, and an F1-score of 0.98204.

Table 4. Model Evaluation Using Held-Out Test Set

Model	Accuracy	Precision	Recall	F1-Score
LR	0.97718	0.98730	0.97718	0.98204
SVM	0.98843	0.99848	0.98843	0.99201
Boosting	0.99786	0.99884	0.99786	0.99833
XGB	0.99954	0.99956	0.99954	0.99955
Bagging	0.99980	0.99981	0.99980	0.99980
KNN	0.99984	0.99986	0.99984	0.99985
Stacking Ensemble Learning	0.99990	0.99990	0.99990	0.99990

The confusion matrix for Stacking Ensemble Learning as in Figure 4, demonstrates its classification performance across all attack categories. The model accurately classified all 77 attack-free instances (class 0) and 682 NTP DDoS attack instances (class 6), with only one misclassification into class 5 (MITM attack). Similarly, DoS attacks (class 1) were correctly classified in 91,381 instances, with only three misclassified as port scan attacks (class 7), while flood attacks (class 2) had 30 correctly classified instances, with one misclassified as port scan attacks (class 7). Fuzzy attacks (class 3) and IEC 104 starvation attacks (class 4) exhibited high accuracy, with only one and two misclassifications, respectively. A minor number of misclassifications were observed, such as one MITM attack (class 5) being classified as NTP DDoS (class 6) and two port scan attacks (class 7) being classified as MITM attacks (class 5).

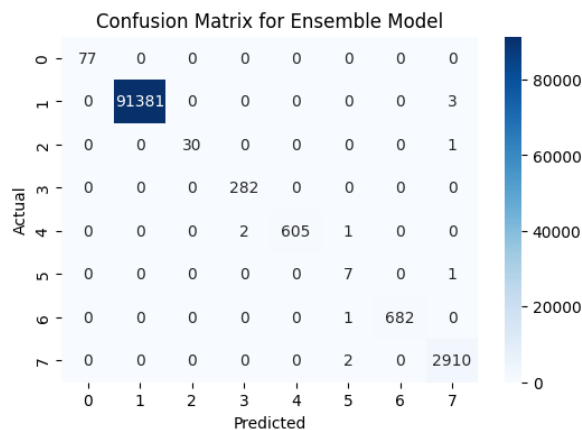


Figure 4. Stacking Ensemble Learning Confusion Matrix

The t-test results, as shown in Table 5, indicate statistically significant differences between the Stacking Ensemble Learning model and the other model. The p-values for LR (0.00428), SVM (0.04237), XGB (0.00000), KNN (0.00057), Bagging (0.00000),

and Boosting (0.00549) are all below the 0.05 threshold, confirming that the Stacking model significantly outperforms these base models. The negative T-statistic values indicate that the Stacking Ensemble model scores higher than individual models. The most substantial difference is observed with XGBoost ($T = -119.51151$, $p = 0.00000$), followed by KNN ($T = -15.58846$, $p = 0.00057$), demonstrating that even high-performing base models are further improved through ensemble learning. The smallest but still significant difference is found with SVM ($T = -3.40280$, $p = 0.04237$), suggesting that while SVM performs well, the stacking approach enhances its performance. Interestingly, Bagging also significantly improves ($T = -322.00000$, $p = 0.00000$), while Boosting shows a notable but slightly smaller difference ($T = -7.21218$, $p = 0.00549$).

Table 5. T-Test Result

Model	Comparison (vs. Stacking Ensemble Learning)	T-Statistic	P-Value
LR		-7.86417	0.00428
SVM		-3.40280	0.04237
XGB		-119.51151	0.00000
KNN		-15.58846	0.00057
Bagging		-322.0	0.00000
Boosting		-7.21218	0.00549

The experimental results demonstrate that the proposed model outperforms all other models evaluated in both the 5-fold CV scenario and testing with held-out data, achieving performance scores of 0.99990. When compared to previous studies, this achievement represents a significant improvement, as shown in Table 6, where the proposed Stacking Ensemble Learning model consistently outperforms other approaches in all evaluation metrics. This superior performance confirms that the Stacking Ensemble Learning method effectively utilizes the complementary strengths of the base models and combines them through an efficient meta-learner, resulting in a better performance model.

The confusion matrix analysis confirms the Stacking Ensemble Learning model's ability to accurately classify a wide range of cyberattack types with minimal misclassifications. This high level of precision is particularly critical for IDS in electrical substations, where false positives can lead to operational disruptions, and false negatives can result in undetected intrusions with potentially severe consequences. The model demonstrated strong class-wise performance, distinguishing between normal traffic and various attack categories.

Table 6. Comparison to Other Studies

Study	Model Used	Accuracy	Precision	Recall	F1-Score
Zhu et al. [12]	RF	0.9939	-	-	-
Shanti et al. [13]	IFM	0.99	-	0.87	0.78
	SVM	0.95	-	0.88	0.67
Sharma and Shah [14]	Hybrid (RF + SVM)	0.9898	0.979	0.976	0.999
Proposed Method	Stacking Ensemble (LR, SVM, KNN, XGB)	0.99990	0.99990	0.99990	0.99990

Furthermore, statistical validation using paired t-tests shows that the performance improvements achieved by the Stacking Ensemble Learning model are statistically significant across all evaluation. All p-values were below the 0.05 significance threshold, indicating that the observed enhancements are unlikely to have occurred by chance. This statistical evidence reinforces the model's superiority over individual base classifiers and other ensemble approaches, such as Bagging and Boosting.

These outcomes demonstrate that the proposed Stacking Ensemble Learning IDS is highly accurate, consistent, and reliable. It is a robust solution well-suited for practical deployment in real-world electrical substation environments, where timely and the detection accuracy of cyber threats is crucial.

4. Conclusions

The evaluation results proven that the proposed Stacking Ensemble Learning significantly outperforms the other models in network intrusion detection with an accuracy of 0.99994, precision of 0.99994, recall of 0.99994, and F1-score of 0.99994 during 5-fold CV and 0.99990, 0.99990, 0.99990, and 0.99990, respectively, on the held-out test set. T-test confirmed the superiority of Stacking Ensemble Learning with all significant p-values below 0.05, indicating a robust improvement in performance. Compared to traditional models and other ensemble techniques such as Bagging and Boosting, Stacking Ensemble Learning effectively combines the strengths of its base models and outperforms them in terms of accuracy and generalization. These findings suggest that Stacking Ensemble Learning is a highly robust approach for electrical substations cybersecurity applications, providing superior classification performance and ensuring enhanced network security. Future work may explore the applications of the proposed model in real-world IDS to assess performance under operational constraints, as well as adaptation for other industrial communication protocols beyond IEC 60870-104, such as Modbus, DNP3, or IEC 61850, to broaden its applicability across various critical infrastructure.

Acknowledgements

The author declares no conflict of interest. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

- [1] M. L. De Klerk and A. K. Saha, "A Review of the Methods Used to Model Traffic Flow in a Substation Communication Network," *IEEE Access*, vol. 8, pp. 204545–204562, 2020, doi: 10.1109/ACCESS.2020.3037143.
- [2] M. F. Elrawy, L. Hadjidemetriou, C. Laoudias, and M. K. Michael, "Light-weight and Robust Network Intrusion Detection for Cyber-attacks in Digital Substations," in *2021 IEEE PES Innovative Smart Grid Technologies - Asia (ISGT Asia)*, IEEE, Dec. 2021, pp. 1–5. doi: 10.1109/ISGTAsia49270.2021.9715626.

- [3] M. A. Aftab, S. M. S. Hussain, I. Ali, and T. S. Ustun, "IEC 61850 based substation automation system: A survey," *International Journal of Electrical Power & Energy Systems*, vol. 120, p. 106008, Sep. 2020, doi: 10.1016/j.ijepes.2020.106008.
- [4] P. Wu, D. Wang, and J. Wu, "IEC60870-5-104 Protocol Security Analysis and Automated Simulation Attack System," in *2024 International Conference on Electronics and Devices, Computational Science (ICEDCS)*, IEEE, Sep. 2024, pp. 336–342. doi: 10.1109/ICEDCS64328.2024.00066.
- [5] X. Wu, G. Qu, H. Jin, and L. Zhang, "Digital Substation Security Monitoring Network Intrusion Detection Based on Deep Learning," in *2023 3rd International Conference on Energy, Power and Electrical Engineering (EPEE)*, IEEE, Sep. 2023, pp. 1091–1094. doi: 10.1109/EPEE59859.2023.10352047.
- [6] R. Zhao et al., "A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things," *IEEE Internet Things J*, vol. 9, no. 12, pp. 9960–9972, Jun. 2022, doi: 10.1109/JIOT.2021.3119055.
- [7] U. Ahmed et al., "Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering," *Scientific Reports* 2025 15:1, vol. 15, no. 1, pp. 1–33, Jan. 2025, doi: 10.1038/s41598-025-85866-7.
- [8] M. A. Umer, K. N. Junejo, M. T. Jilani, and A. P. Mathur, "Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations," *International Journal of Critical Infrastructure Protection*, vol. 38, p. 100516, Sep. 2022, doi: 10.1016/j.ijcip.2022.100516.
- [9] A. Vikram and Mohana, "Anomaly detection in Network Traffic Using Unsupervised Machine learning Approach," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, Jun. 2020, pp. 476–479. doi: 10.1109/ICCES48766.2020.9137987.
- [10] S. A. Bakhsh, M. A. Khan, F. Ahmed, M. S. Alshehri, H. Ali, and J. Ahmad, "Enhancing IoT network security through deep learning-powered Intrusion Detection System," *Internet of Things*, vol. 24, p. 100936, Dec. 2023, doi: 10.1016/j.iot.2023.100936.
- [11] A. Verma and V. Ranga, "Machine Learning Based Intrusion Detection Systems for IoT Applications," *Wirel Pers Commun*, vol. 111, no. 4, pp. 2287–2310, Apr. 2020, doi: 10.1007/s11277-019-06986-8.
- [12] G. ZHU, H. YUAN, Y. ZHUANG, Y. GUO, X. ZHANG, and S. QIU, "Research on network intrusion detection method of power system based on random forest algorithm," in *2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, IEEE, Jan. 2021, pp. 374–379. doi: 10.1109/ICMTMA52658.2021.00087.
- [13] K. Shanthi and R. Maruthi, "Machine Learning Approach for Anomaly-Based Intrusion Detection Systems Using Isolation Forest Model and Support Vector Machine," in *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, IEEE, Aug. 2023, pp. 136–139. doi: 10.1109/ICIRCA57980.2023.10220620.
- [14] V. Sharma and D. J. Shah, "A Novel Approach to Intrusion Detection Systems Using Hybrid Machine Learning Techniques," in *2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAQSA)*, IEEE, Dec. 2024, pp. 1–6. doi: 10.1109/ICAQSA64000.2024.10882184.
- [15] L. Yang and A. Shami, "IDS-ML: An open source code for Intrusion Detection System development using Machine Learning," *Software Impacts*, vol. 14, p. 100446, Dec. 2022, doi: 10.1016/j.simpa.2022.100446.
- [16] E. Retnoningsih and R. Pramudita, "Mengenai Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python," *BINA INSANI ICT JOURNAL*, vol. 7, no. 2, p. 156, Dec. 2020, doi: 10.51211/biict.v7i2.1422.
- [17] R. Xu, G. Wu, W. Wang, X. Gao, A. He, and Z. Zhang, "Applying self-supervised learning to network intrusion detection for network flows with graph neural network," *Computer Networks*, vol. 248, p. 110495, Jun. 2024, doi: 10.1016/j.comnet.2024.110495.

- [18] Ajeng Arifa Chantika Rindu, Ria Astriratma, and Ati Zaidiah, "K-Means Algorithm Implementation for Project Health Clustering," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 5, pp. 1064–1076, Sep. 2023, doi: 10.29207/resti.v7i5.5181.
- [19] Md. A. Talukder et al., "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," *J Big Data*, vol. 11, no. 1, p. 33, Feb. 2024, doi: 10.1186/s40537-024-00886-w.
- [20] X. Du and W. Zhao, "Risky lane-changing behavior recognition based on stacking ensemble learning on snowy and icy surfaces," *Sci Rep*, vol. 14, no. 1, p. 19257, Aug. 2024, doi: 10.1038/s41598-024-00842-7.
- [21] U. Ahmed et al., "Explainable AI-based innovative hybrid ensemble model for intrusion detection," *Journal of Cloud Computing*, vol. 13, no. 1, p. 150, Oct. 2024, doi: 10.1186/s13677-024-00712-x.
- [22] E. D. Gutiérrez Mlot, J. Saldana, R. J. Rodríguez, I. Kotsiuba, and C. Gañán, "A dataset to train intrusion detection systems based on machine learning models for electrical substations," *Data Brief*, vol. 57, p. 111153, Dec. 2024, doi: 10.1016/J.DIB.2024.111153.
- [23] A. Gumilar, S. S. Prasetyowati, and Y. Sibaroni, "Performance Analysis of Hybrid Machine Learning Methods on Imbalanced Data (Rainfall Classification)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 3, pp. 481–490, Jul. 2022, doi: 10.29207/resti.v6i3.4142.
- [24] Y. Prihantono and Kalamullah Ramli, "Model-Based Feature Selection for Developing Network Attack Detection and Alerting System," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 2, pp. 322–329, Apr. 2022, doi: 10.29207/resti.v6i2.3989.
- [25] R. Shaikh, M. Rafi, N. A. Mahoto, A. Sulaiman, and A. Shaikh, "A filter-based feature selection approach in multilabel classification," *Mach Learn Sci Technol*, vol. 4, no. 4, p. 045018, Dec. 2023, doi: 10.1088/2632-2153/ad035d.
- [26] Nia Mauliza, Aisha Shakila Iedwan, Yoga Pristyanto, Anggit Dwi Hartanto, and Arif Nur Rohman, "The Effect of Resampling Techniques on Model Performance Classification of Maternal Health Risks," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, no. 4, pp. 496–505, Aug. 2024, doi: 10.29207/resti.v8i4.5934.
- [27] P. Ghose, K. Oliullah, M. K. Mahbub, M. Biswas, K. N. Uddin, and H. M. Jamil, "Explainable AI assisted heart disease diagnosis through effective feature engineering and stacked ensemble learning," *Expert Syst Appl*, vol. 265, p. 125928, Mar. 2025, doi: 10.1016/j.eswa.2024.125928.
- [28] B. Sunarko et al., "Penerapan Stacking Ensemble Learning untuk Klasifikasi Efek Kesehatan Akibat Pencemaran Udara," *Edu Komputika Journal*, vol. 10, no. 1, pp. 55–63, Sep. 2023, doi: 10.15294/edukomputika.v10i1.72080.
- [29] R. Kalule, H. A. Abderrahmane, W. Alameri, and M. Sassi, "Stacked ensemble machine learning for porosity and absolute permeability prediction of carbonate rock plugs," *Sci Rep*, vol. 13, no. 1, p. 9855, Jun. 2023, doi: 10.1038/s41598-023-36096-2.
- [30] D. M. A. S. Elkahwagy, C. J. Kiriacos, and M. Mansour, "Logistic regression and other statistical tools in diagnostic biomarker studies," *Clinical and Translational Oncology*, vol. 26, no. 9, pp. 2172–2180, Mar. 2024, doi: 10.1007/s12094-024-03413-8.
- [31] W. Zhao and Z. Zhao, "Providing a hybrid approach to increase the accuracy of intrusion detection systems in computer networks," *Journal of Engineering and Applied Science*, vol. 71, no. 1, p. 123, Dec. 2024, doi: 10.1186/s44147-024-00404-y.
- [32] Agus Supriyanto, R. Rizal Isnanto, and Oky Dwi Nurhayati, "Klasifikasi Penyakit Daun Kopi Robusta Menggunakan Metode SVM dengan Ekstraksi Ciri GLCM," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 12, no. 4, pp. 241–248, Nov. 2023, doi: 10.22146/jnteti.v12i4.8044.
- [33] Wikky Fawwaz Al Maki, Khaidir Mauladan, and Indra Bayu Muktyas, "Date Fruit Classification using K-Nearest Neighbor with Principal Component Analysis and Binary Particle Swarm Optimization," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 6, pp. 1456–1463, Dec. 2023, doi: 10.29207/resti.v7i6.4839.
- [34] S. K. Kiangala and Z. Wang, "An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment," *Machine Learning with Applications*, vol. 4, p. 100024, Jun. 2021, doi: 10.1016/j.mlwa.2021.100024.
- [35] Erfian Junianto, Mila Puspitasari, Salman Ilyas Zakaria, Toni Arifin, and Ignatius Wiseto Prasetyo Agung, "Klasifikasi Emosi pada Teks Berbahasa Inggris Menggunakan Pendekatan Ensemble Bagging," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 13, no. 4, pp. 272–281, Nov. 2024, doi: 10.22146/jnteti.v13i4.14440.
- [36] K. Soman, C. Kokate, A. Mohite, A. Vispute, O. More, and Prof. Z. K. Mundargi, "Statistical Tests for Comparing Machine Learning Algorithms," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 12, pp. 628–633, Dec. 2022, doi: 10.22214/ijraset.2022.47955.