Intrusion Detection for Power System Security by Ensemble Learning with Auxiliary Classifier and Feature Selection

SI-Wei Lee[1] and Jen-Yeu Chen[1]

[1]Department of Electrical Engineering, National Dong Hwa University Hualien

April 09, 2024

Abstract

This paper proposes a stacking framework based on ensemble learning, aiming to establish a machine learning-based intrusion detection system to accurately differentiate various cyber-attack types that pose security risks to substations. The framework utilizes a combination of stacked base learners and secondary learners to generate binary feature matrices based on the probability weighting of natural or attack events and multi-class feature matrices of the probability of occurrence of all attack events. The model designed in this paper is trained using the power system attack detection dataset developed by the Oak Ridge National Laboratory at Mississippi State University. In the experimental results, the binary classification accuracy of the secondary learner reaches 97%, and the multiclass accuracy reaches 95%. This paper also discusses the importance of feature selection techniques for intrusion detection systems. Experimental results show that using RFE can maintain the model's accuracy at around 95% across different training/test set ratios of 9:1, 8:2, and 7:3.

Hosted file

`Equation.docx` available at https://authorea.com/users/765143/articles/767387-intrusion-detection-for-power-system-security-by-ensemble-learning-with-auxiliary-classifier-and-feature-selection

Hosted file

`Table.docx` available at https://authorea.com/users/765143/articles/767387-intrusion-detection-for-power-system-security-by-ensemble-learning-with-auxiliary-classifier-and-feature-selection

Hosted file

`Graph.docx` available at https://authorea.com/users/765143/articles/767387-intrusion-detection-for-power-system-security-by-ensemble-learning-with-auxiliary-classifier-and-feature-selection

Intrusion Detection for Power System Security by Ensemble Learning with Auxiliary Classifier and Feature Selection

Si-Wei Lee, Jen-Yeu Chen
Department of Electrical Engineering National Dong Hwa University
Hualien, Taiwan, R.O.C.
{611123025, jenyeu}@gms.ndhu.edu.tw

***Abstract* — This paper proposes a stacking framework based on ensemble learning, aiming to establish a machine learning-based intrusion detection system to accurately differentiate various cyber-attack types that pose security risks to substations. The framework utilizes a combination of stacked base learners and secondary learners to generate binary feature matrices based on the probability weighting of natural or attack events and multi-class feature matrices of the probability of occurrence of all attack events. The model designed in this paper is trained using the power system attack detection dataset developed by the Oak Ridge National Laboratory at Mississippi State University. In the experimental results, the binary classification accuracy of the secondary learner reaches 97%, and the multi-class accuracy reaches 95%. This paper also discusses the importance of feature selection techniques for intrusion detection systems. Experimental results show that using RFE can maintain the model's accuracy at around 95% across different training/test set ratios of 9:1, 8:2, and 7:3.**

Index Terms—Cyber-attack, Ensemble learning, Intrusion detection system(IDS), Power system, Stacking.

I. INTRODUCTION

THE power system, as an indispensable cornerstone of modern society, plays a crucial role in ensuring our quality of life and economic operations. The stable operation of the power system is directly related to these aspects. In this era of high digitization and widespread internet connectivity, the power system not only needs to meet the continuously growing demand for electricity but also must confront complex security challenges.

With technological advancements, the power system faces risks not only from natural disasters but also from new threats such as cyberattacks and data injection. In recent years, security issues related to power systems have been escalating [1], and the need for information security in power systems requires urgent attention. This paper designs an intrusion detection system based on ensemble learning, employing stacking as the intrusion detection framework.

This paper designs an intrusion detection system based on ensemble learning, employing stacking [2] as the intrusion detection framework. The binary and multiclass labels from the Oak Ridge National Laboratory (ORNL) power system attack detection dataset[3] developed by Mississippi State University are merged into a new dataset.

This merging aims to facilitate the training process for the secondary learner, which employs a multilayer perceptron along with an additional auxiliary classifier. The design allows the secondary learner to simultaneously obtain binary classification results distinguishing between attacks and natural events and multiclass results providing a detailed classification of attack event types during the training process.

The rest of this paper is organized as follows. Selection II introduces the architecture and workflow diagram of the proposed intrusion detection system. Selection III provides an overview of the base learners and the secondary learner used in this study. Selection IV describes the ORNL power system dataset. Selection V outlines the experimental setup and simulation results. Finally, Selection VI concludes the study.

II. THE FRAMEWORK OF THE INTRUSION DETECTION SYSTEM

The intrusion detection system framework proposed by us leverages a stacking approach to enhance overall classification performance.

A. Framework in IDS

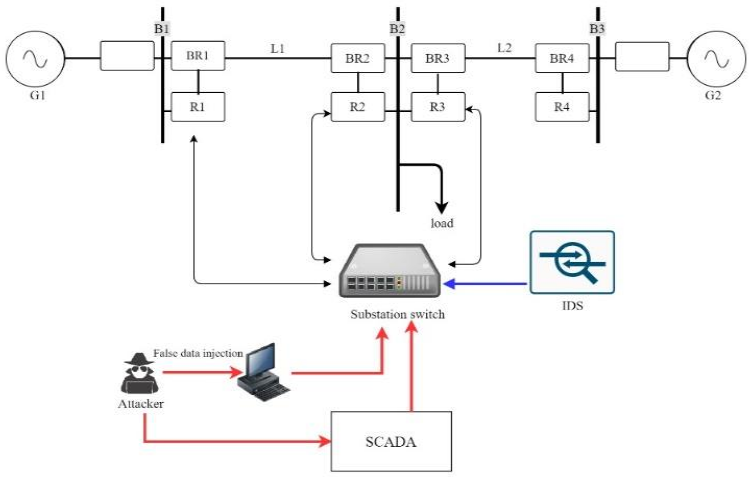

Fig. 1. Intrusion Detection Framework for Power Systems

Our proposed intrusion detection system framework is depicted in Fig. 1 The topmost section illustrates the power system architecture, modeled after the two-generator three-bus system from the ORNL dataset [3]. This architecture simulates the entire process of a small-scale power system, including Generation (G1 & G2), which simulate power generation within the system, and four PMUs (Phase Measurement Units), which analogously represent components found in a substation and are used to simulate distribution behavior. Additionally, L1 and L2 simulate transmission behavior.

Fig. 1 depicts four PMUs (including four relays R1-R4 and four circuit breakers BR1-BR4) as switches within the substation, with the intrusion detection system(IDS) placed near these switches. Attackers typically send malicious attacks via a PC and manipulate data on the transmission lines to the substation switches. Another pathway involves sending false data to the substation via the Supervisory Control and Data Acquisition (SCADA) system. SCADA ensures the security of the power system by monitoring the grid, sampling data (such as voltage, current, frequency, etc.), controlling equipment (such as circuit breakers, switches, etc.), and providing real-time messages and historical data.

B. *IDS Architecture*

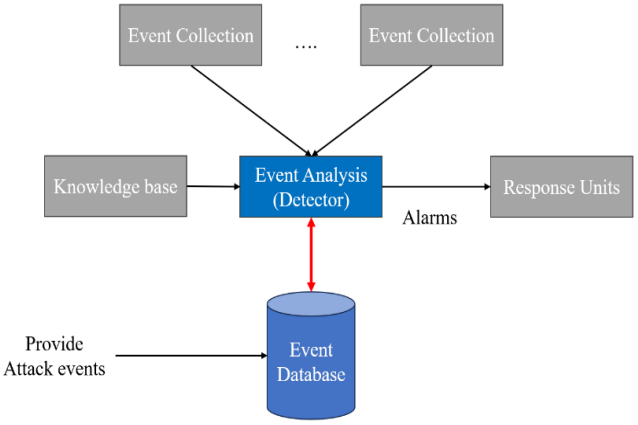

Fig. 2 Classical Intrusion Detection System

[4] proposed a basic intrusion detection system (IDS) architecture with slight modifications. Fig. 2 of the IDS primarily includes the following components:

1) *Event Collection:* The event generator (or event collector) is responsible for monitoring activities within the power system, such as voltage and current variations, equipment connections, and disconnections.
2) *Event Analyzer:* The event analyzer is tasked with analyzing the data collected by the event generator to identify potential security risks. The machine learning model designed in this paper is deployed within the event analyzer to facilitate the detection of unknown threats and attacks.
3) *Event Database:* The event database stores data on power system events, including both natural occurrences and attacks. The ORNL dataset used in this paper documents a wealth of natural and attack events in power systems to improve the performance of the IDS.
4) *Response Units:* Response units are devices that take appropriate actions based on the results of event analysis. In the power system designed in this paper, if the machine learning model within the event analyzer identifies an attack event, it will generate an alert and transmit it to the response units. For example, relays may send commands to circuit breakers, which then receive the commands to further disconnect and protect the entire system.

III. ORNL DATASET

The electric power system dataset used in this paper is sourced from a public dataset provided by the Oak Ridge National Laboratory (ORNL) [3]. Nowadays, many studies utilize this dataset for research on intrusion detection systems, with applications ranging from network attack detection to power system IDS and simulated smart grid attack detection.

A. *Data Labeling*

The ORNL dataset comprises uninterrupted measurements of voltage and other power-related data by PMUs. It contains approximately 70,000 samples, which are divided into 15 datasets with around 5,000 samples each. These datasets are labeled based on scenarios, categorizing them into Binary, Three-class, and Multiclass. The measurement data in the 15 datasets with three types of labels are the same; the difference lies only in the number of labels.

B. *Dataset Features*

Each of the 15 datasets contains 128 features and 1 label. In this experiment, the model used needs to classify whether the event belongs to an attack or a natural scene before training, so an additional binary label "$attack\_scene$" is added (1 represents an attack event, 0 represents a natural event), resulting in a total of 130 features.

TABLE I
DESCRIPTION OF PMU MEASUREMENT FEATURES IN ORNL DATASET

Feature	Unit	Describe
PA1:VH~PA3:VH	rad	The voltage phase angles of the Phase A,B,C electrical system
PA4:IH~PA6:IH	rad	The current phase angles of the Phase A,B,C electrical system
PA7:VH~PA9:VH	rad	The voltage phase angles of the unbalanced Phase A, B, C (positive, negative, zero)
PA10:IH~PA12:IH	rad	The current phase angles of the unbalanced Phase A, B C (positive, negative, zero)
PM1:V~PM3:V	V	The voltage magnitude of the Phase A, B, C electrical system
PM4:I~PM6:I	A	The current magnitude of the Phase A, B, C electrical system
PM7:V~PM9:V	V	The voltage magnitude of the unbalanced Phase A, B, C (positive, negative, zero)
PM10:I~PM12:I	A	The current magnitude of the unbalanced Phase A, B, C (positive, negative, zero)
F	Hz	The frequency of a power system measured by a PMU
DF	Hz/s	Power system frequency variation rate
PA:Z	Ω	PMU-measured equivalent impedance
PA:ZH	degree	Phase angle difference of PMU-measured equivalent impedance
S		The status flag of a PMU

The features starting with "PA" indicate phase angles measured by four PMUs (R1~R4, refer to Fig. 1), with each PMU measuring 29 data points (29*4 = 116). The remaining 12 features belong to the status logs. Finally, the last two

labels are used to determine the scene "*marker*" and classify whether it belongs to an attack or natural scene "*attack_scene*".

C. Event Scenarios

TABLE II
DESCRIPTION OF NATURAL EVENTS AND NO-EVENT IN ORNL DATASET

Scenario number	Recode	Description
1	0	A SLG fault originating from the 10-19% location of L1.
2	0	A SLG fault originating from the 20-79% location of L1.
3	0	A SLG fault originating from the 80-90% location of L1.
4	0	A SLG fault originating from the 10-19% location of L2.
5	0	A SLG fault originating from the 20-79% location of L2.
6	0	A SLG fault originating from the 80-90% location of L2.
13	0	Line maintenance on L1.
14	0	Line maintenance on L2.
41	0	Normal operation load variation.

TABLE III
DESCRIPTION OF ATTACK EVENTS IN ORNL DATASET

Scenario number	Recode	Description
7~12	1~6	False date injection attacks
15~20	7~12	Remote tripping command injection attacks
21~40	13~28	Relay setting changed attacks

The event scenes are labels in the dataset, displayed in numerical order from 1 to 41 (with numbers 31 to 34 missing), totaling 37 scenes. These scenes include 8 natural events, 1 no event, and 28 attack events. Table II and Table III list the scene numbers and descriptions for natural events, no events, and attack events.

In Table II, all natural events and no-event scenarios are coded as zero in this paper's encoding, as the focus is specifically on classifying attack events. In Table III , all attack events are renumbered from 1 to 28 in sequence in this paper's encoding. This renumbering applies to "*marker*." For "*attack_scene*," values from 1 to 28 are set to 1, while the rest are set to zero.

IV. STACKING MODEL

This section will introduce the machine learning model architecture used in this paper, including the base learners and the meta-learner.

A. Architecture

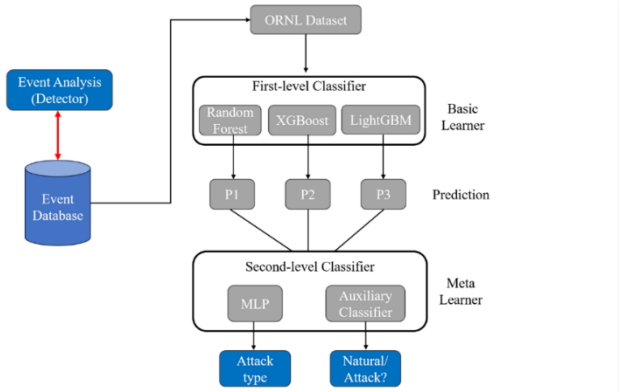

Fig. 3. Model Architecture Based on Stacking

The event analyzer of the IDS corresponds to the machine learning model used in this paper, while the event database corresponds to the ORNL dataset. The event analyzer can utilize machine learning techniques to learn normal patterns and identify abnormal behavior. Once the event analyzer detects anomalies, it may generate alerts.

These alerts are typically sent to response units such as relays and circuit breakers. Upon receiving the alerts, response units can take appropriate actions, such as disconnecting the affected circuit to prevent further damage or security risks.

The overall model workflow begins with the input of attack and natural events from the ORNL dataset. The dataset is then separately used as input for three base learners, resulting in predictions from three machine learning models. At this stage, there are a total of three outputs for binary classification (natural or attack events) and three outputs for multiclass classification (1-29 attack events) matrices. Combining the predictions from these three base classifiers yields a 93 (2*3 + 29*3) matrix as input for the meta-learner. The meta-learner's auxiliary head provides preliminary classification of natural or attack events, while the output layer provides detailed classification of attack events.

B. Basic Learner

Stacking involves using the predictions of multiple base learners (such as Random Forest, AdaBoost, XGBoost, etc.) as inputs and using a secondary learner (often a linear regressor or another model) to integrate these predictions. There are no restrictions on the choice of base learners and the secondary learner. In this paper, the selection of base learners is based on their accuracy in training with the power system ORNL dataset.

TABLE IV
THE SELECTION OF BASIC LEARNERS FOR STACKING

ML algorithm	Binary Accuracy	Multiple Accuracy
Decision Tree	0.7	0.31
SVM	0.69	0.29
Naïve Bayse	0.69	0.28
Random Forest	0.92	0.8
Adaboost	0.68	0.29
XGBoost	0.71	0.68
LightGBM	0.77	0.71

Table IV shows the training results of different machine learning algorithms using the original unencoded ORNL dataset. Based on the comparison of accuracy, we selected Random Forest, XGBoost, and LightGBM as the base learners for the first layer of the stacking model.

C. Meta Learner

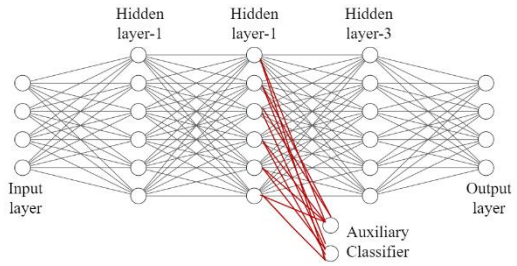

Fig. 4. The Architecture of the Meta Learner in Stacking

The diagram in Fig. 4 illustrates the secondary learner of the stacking model, which in this study is a Multi-Layer Perceptron (MLP).

The architecture in Fig. 4 is primarily divided into four layers: the input layer, hidden layers, auxiliary layer, and output layer. The input to the input layer is the predictions obtained from the three basic learners' training results, which are merged into a feature matrix of size (training samples * (number of multi-class/binary labels * 3)).

After multiple feature extractions between hidden layers through forward propagation, there are two pathways: the auxiliary layer and the output layer. Backward propagation occurs through these two pathways back to the input layer.

The concept of the Auxiliary Classifier (AC) was first introduced in 2014 by the Google team in the development of GoogleNet (Inception-V1) [5]. Since GoogleNet is a large neural network consisting of multiple layers of fully connected (FC) layers, convolutional layers, and an average pooling layer, it may suffer from the problem of gradient vanishing during backpropagation, leading to ineffective updates in the earlier layers and reduced performance.

The AC is typically placed in the middle layers of the model and calculates the output of these middle layers. This positioning allows the AC to be shielded from direct gradient vanishing issues, even if there are many hidden layers between the output and input layers. This is because during backpropagation, the AC is in the middle layers, making the gradients less likely to vanish directly to zero.

The definition of the loss function for the main classifier can be written as:

$$L_{main} = \sum_{i=1}^{n}\sum_{i=1}^{c} y_{c,i}\log_2(P_{c,i}) \tag{1}$$

where n represents the number of samples, c represents the type of attack category in the ORNL dataset, $y_{c,i}$ is a binary indicator used for one-hot encoding, and $P_{c,i}$ is the probability that the ith sample is classified as category c.

The definition of the loss function for the auxiliary classifier can be written as:

$$L_{aux} = \sum_{i=1}^{n}\sum_{i=1}^{c} y_{c,i}\log_2(P_{c,i}) \tag{2}$$

The difference with the loss function of the main classifier lies in the fact that c represents a binary indicator (0, 1) for determining whether it belongs to a natural or attack scene.

V. Simulaton Results

In this chapter, we first followed the workflow of our work and then presented the experimental results. Additionally, we compare the performance with other papers that have utilized the ORNL dataset.

A. Workflow

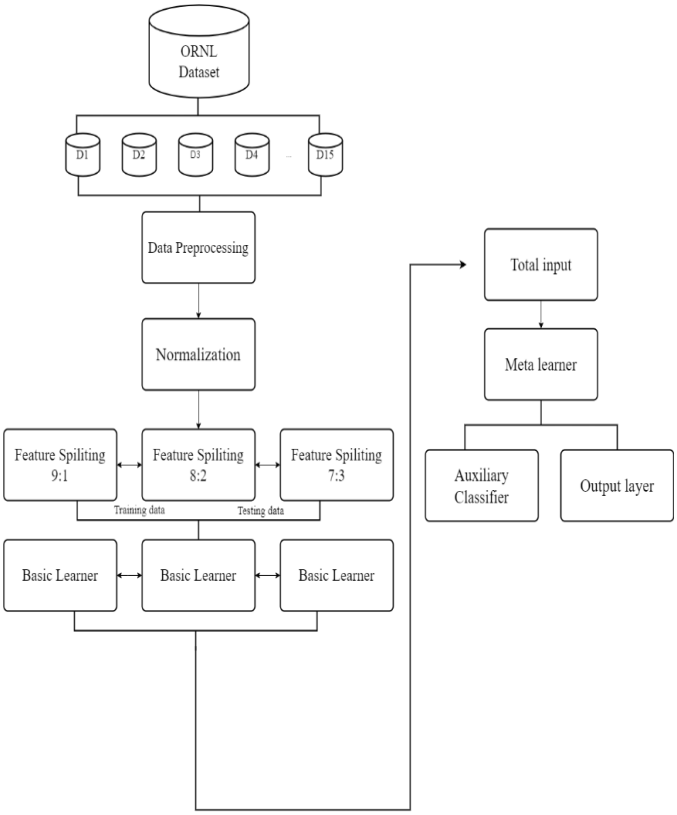

Fig. 5. The workflow in proposed IDS model.

1) *Data preprocessing:* Data preprocessing involves cleaning the power-related measurements collected by PMUs. In this study, missing values in the measurements are handled by directly removing them.
2) *Normalization:* In a three-phase power system, the values of phase voltages and phase currents measured by PMUs can vary significantly. Therefore, it is necessary to normalize the data in the dataset to ensure that the model can handle consistent data during both training and testing phases. The data normalization method used in this paper is Min-Max

Scaling. This method scales the dataset so that each feature corresponds to the range between 0 and 1 by identifying the maximum and minimum values for each feature in the dataset.

$$X_{nor} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3}$$

where X represents a certain feature of the ORNL dataset, X_{max} denotes the maximum value of that feature, and X_{min} represents the minimum value of that feature.

3) *Data Splitting:* The normalized dataset is then split into training and testing sets according to three different split ratios as depicted in Fig. 5. It is important to note that during the splitting process, there are binary and multiclass labels, resulting in two outputs.

4) *Basic Learner & Combine Input:* After splitting, the training and testing sets are used as inputs for the three base learners. Under the stacking model framework, the predictions from the three base learners are combined as probability values. Utilizing probability values as inputs for the secondary learner provides more information, which is beneficial for classification tasks.

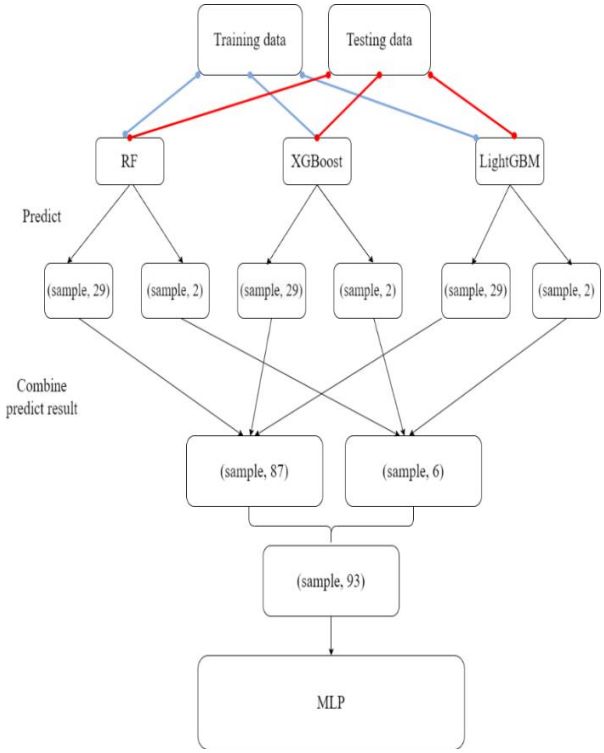

Fig. 6. Stacking model framework diagram.

Fig. 6 elaborates on the combination of outputs from Basic Learners as the input for the Secondary Learner, as depicted in Fig. 4.

The "Predict" step in Fig. 6 involves predicting the binary and multiclass labels in the dataset, resulting in four different-sized feature matrices: $(Training\ sample, 2)$, $(Training\ sample, 29)$, $(Testing\ sample, 2)$, and $(Testing\ sample, 29)$.

The reason for having 29 multiclass labels is that the original dataset's multiclass labels were encoded by numbering all 41 event scenarios from 1 to 41. However, for this IDS in our paper, the focus is primarily on attack events. Therefore, all labels except for attack events are assigned as 0.

After merging the feature matrices based on binary and multiclass labels separately, these two feature matrices are then horizontally concatenated to form a large feature matrix, which serves as the input for the secondary learner.

Probability	Binary feature matrix	
	0	1
	0.34	0.66
	0.7	0.3

Fig. 7. Illustration of Binary Feature Matrix

Probability	Multiclass feature matrix				
	0	1	2	...	28
	0.05	0.07	0.75		0.13
	0.3	0.6	0.02		0.001

Fig. 8. Illustration of Multiple Feature Matrix

Fig.7 and Fig. 8 depict the binary and multiclass feature matrices mentioned above. The model outputs prediction probabilities for each label based on the number of labels. A higher prediction probability for a feature indicates a higher probability of the model assigning that feature to a certain category, which can also be interpreted as the importance level of the feature. The size of the feature matrix is (number of samples, number of features), and the sum of probabilities for each column equals 1.

5) *Meta learner:* Finally, the merged feature matrix is fed into the secondary learner, where a Multilayer Perceptron (MLP) further explores the relationships between features within hidden layers. Based on the architecture depicted in Fig. 4, the auxiliary classifier predicts whether the event belongs to a natural occurrence or an attack. The output layer further categorizes the event into specific types of attacks.

B. *Experiment 1: IDS accuracy*

Experiment 1 focuses on comparing the accuracy of the model trained in this study with the one in reference [6]. Since the training-to-testing data ratio in [6] is 9:1, this study compares the accuracy using three different ratios: 9:1, 8:2, and 7:3.

1) *Compare with the Original Dataset:* The experiment compared the performance using the original dataset without renumbering, as described in Table II and Table III. The results showed that even without renumbering, the model maintained a higher level of accuracy.

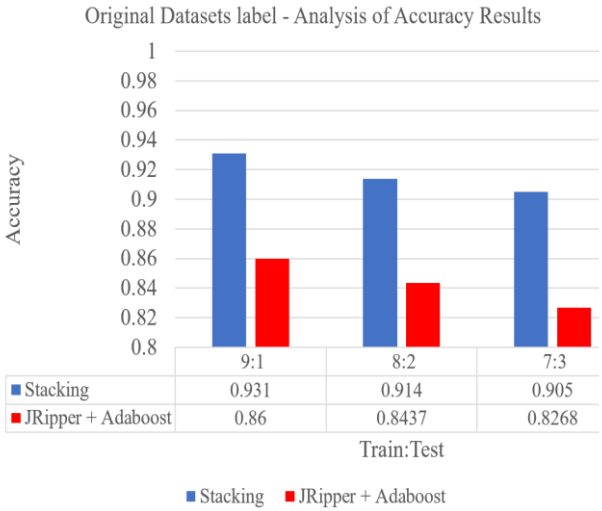

Fig. 9. The comparison of results using the original dataset.

The experimental results using the original data as input with our stacking model are depicted in Fig. 9, showing that our research method maintains an advantage regardless of the amount of training data. Since the original data was used as input, the stacking model in this part does not include an auxiliary classifier, and the legend for models without the auxiliary classifier is labeled as "Stacking." The blue labels in Fig. 9 represent the average accuracy across the 15 datasets, while the red labels denote the experimental results from reference [6].

2) *The comparison of results from after re-coding data:* The accuracy of the 15 datasets using the stacking model's simulation results is recorded in this paper. A comparison is made between training and testing set ratios of 9:1 to 7:3. The goal is to maintain excellent discrimination results even with a small amount of training data.

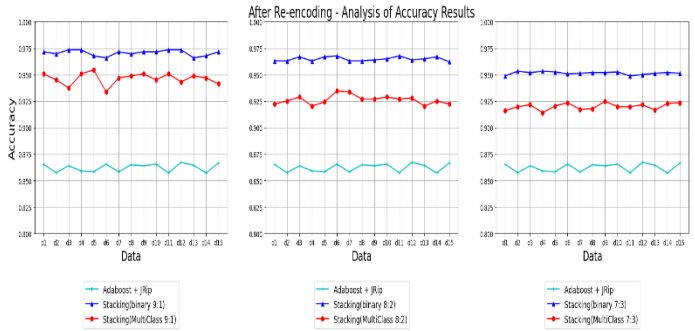

Fig. 10. The comparison of the accuracy using ORNL dataset after re-encoding.

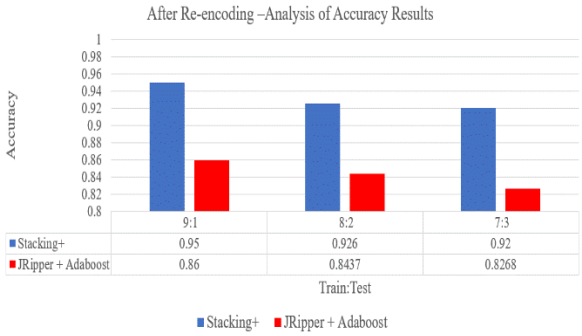

Fig. 11. The comparison of three training and testing set proportions after renumbering.

The accuracy of the 15 datasets using the stacking model's simulation results is recorded in this paper. A comparison is made between training and testing set ratios of 9:1 to 7:3. The goal is to maintain excellent discrimination results even with a small amount of training data.

The results averaged from the 15 datasets in Fig. 10, compared to Fig. 6, show an improvement in accuracy after adding the auxiliary classifier and re-encoding. "Stacking+" in Fig. 11 represents the model with re-encoding and the inclusion of the auxiliary classifier.

C. *Experiment 2: Other Machine Learning Metrics in Proposed IDS*

The data tested in Experiment 2 compares the indicators of recall, precision, and f1-score for proposed IDS.

$$Recall = \frac{TP}{TP + FP} \tag{4}$$

According to the definition of Recall (4), applied in IDS, it represents the system's desire to detect as many true attack events as possible, reducing the risk of IDS false negatives. False negatives indicate cases where the IDS fails to detect certain true attack events.

$$Precision = \frac{TP}{TP + FN} \tag{5}$$

According to the definition of Precision (5), applied in IDS, it represents the system's accuracy in identifying attack

events. High Precision indicates that the IDS predicts the events marked as attacks in the dataset very accurately, reducing the risk of false positives.

$$f1-score = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall}, \qquad \beta = 1 \tag{6}$$

The f1-score (6) is the weighted average between Recall and Precision, applied in situations where there is class imbalance in the dataset. When the number of attack event samples is smaller than that of natural event samples in the dataset, it provides a comprehensive evaluation.

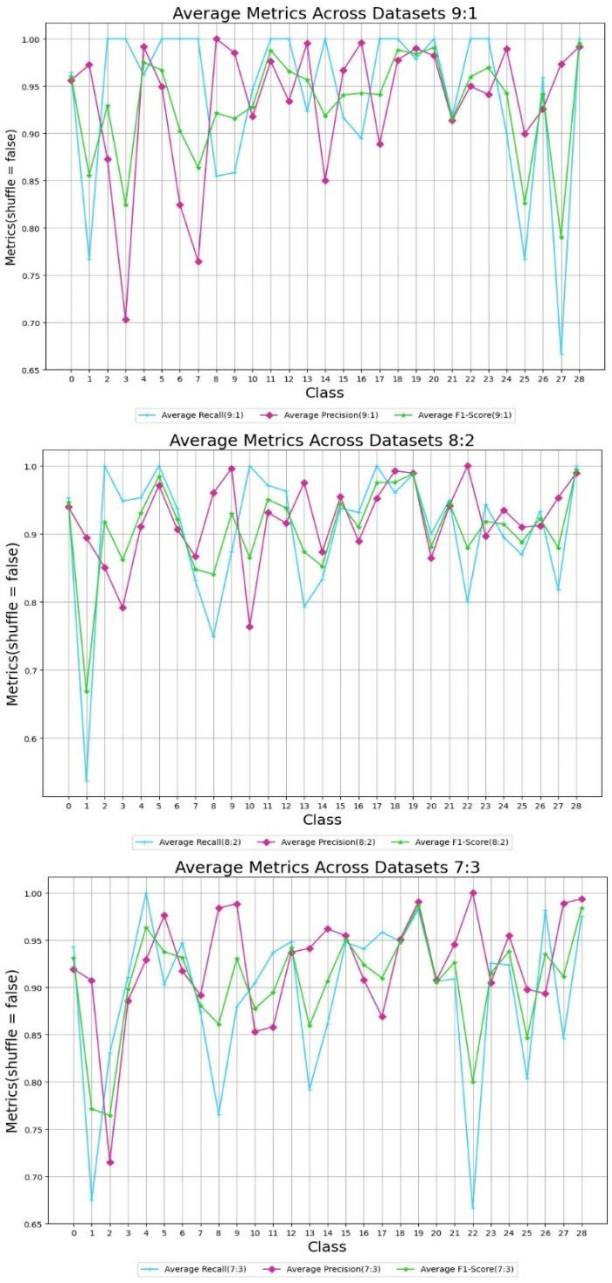

Fig. 12. The comparison of metrics without shuffle

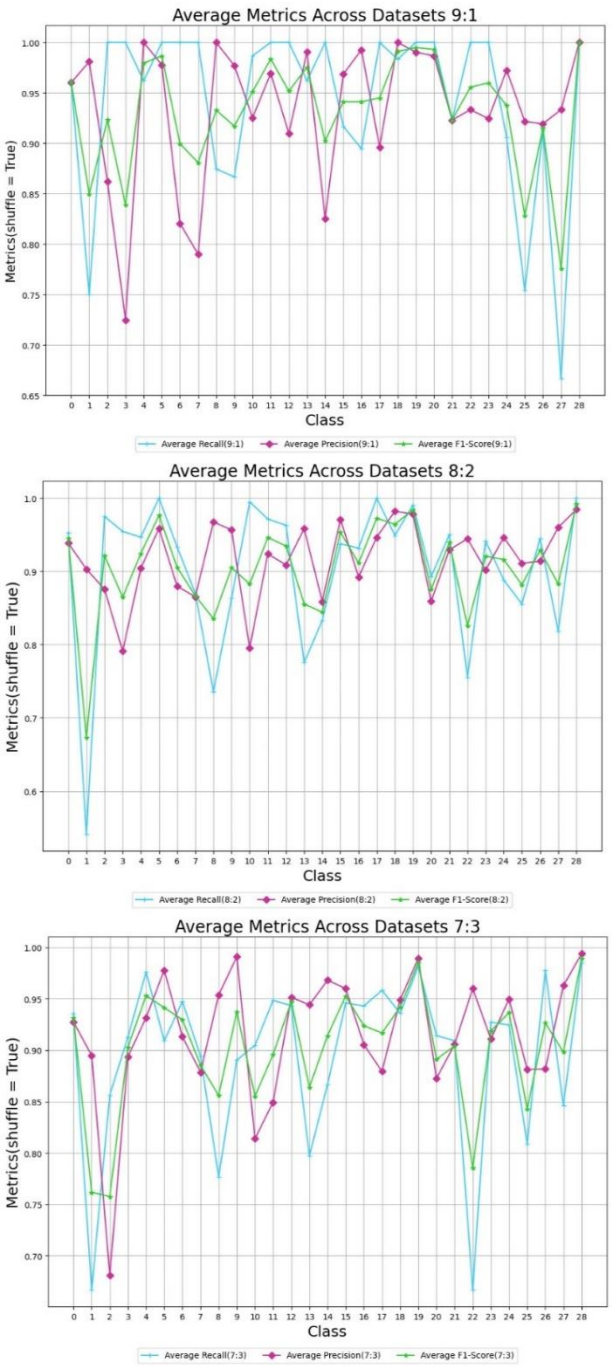

Fig. 13. The comparison of metrics with shuffle

Fig. 12 and Fig. 13 compares machine learning metrics in our study. The Y-axis shows event scene numbers post re-encoding, while the X-axis displays metric averages across the 15 datasets. "Shuffle" refers to rearranging training data for performance improvement.

The plot aims to assess if shuffling affects our IDS model's performance per attack event. Results indicate consistent performance regardless of shuffling, maintaining accuracy across different training set sizes.

Notably, events 2, 3, 6, and 7 show lower Precision, likely due to their nature as data injection attacks, which are challenging to distinguish accurately given their similarity to normal operations.

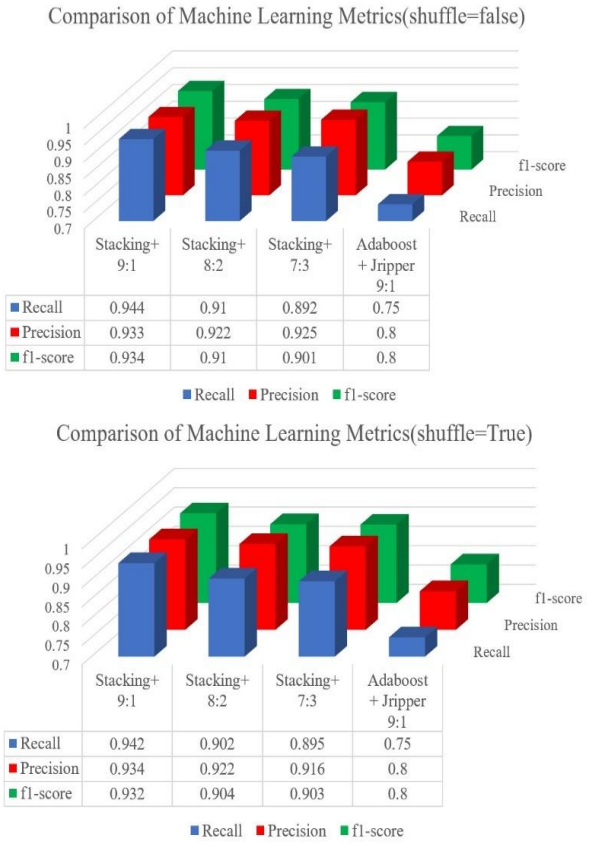

Fig. 14. The comparison of metrics with shuffling in machine learning

Fig. 14 compare the quantized results from Fig. 12 & 13. The comparison shows that shuffling has minimal impact on the metrics. Adaboost+JRipper represents the average results from [6], highlighting significant differences between our IDS model's metrics and those from [6]. The averaging method involves dividing the three metrics of labels 0 to 28 by 29 to obtain the results.

D. Experiment 3: Feature Selection

From Fig. 11, it can be observed that as the size of the training set varies, there is a gradual decrease in accuracy by 2-3%, which is a relatively significant change. Such fluctuations in accuracy indicate insufficient stability in the predictions made by our IDS model. Therefore, experiment 3 involves the application of several feature selection techniques for preprocessing, aiming to reduce the magnitude of accuracy decline with changes in training data size.

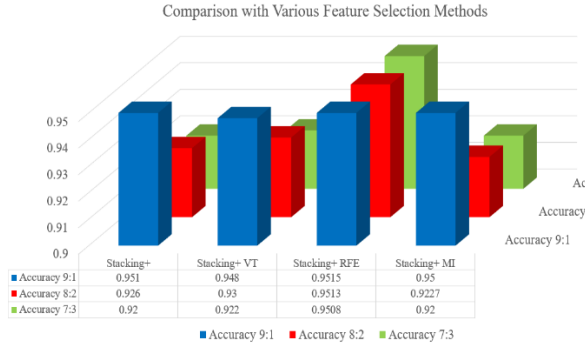

Fig. 15. Feature selection accuracy quantification comparison

The accuracy comparison obtained from training using the intrusion detection model designed in this paper shows that Recursive Feature Elimination (RFE) can maintain the model accuracy around 95% regardless of the amount of training data.

RFE sorts the features based on their importance and then eliminates the least important features iteratively until the desired number of features set by the user is reached. This result may be attributed to RFE capturing the important features for the ORNL dataset during training, and these features are sufficient to represent the data and its overall structure. Hence, RFE can maintain good accuracy even with different amounts of training samples.

E. The Comparsion of Performance with Using ORNL Dataset Reference

TABLE V
THE COMPARISON OF PERFORMANCE WITH PROPOSED MODEL

Research Method	Stacking+	Stacking+ & RFE		
Data Instance	**5,000**	**5,000**		
Train-Test Split	**9:1**	**9:1**	**8:2**	**7:3**
Binary Accuracy	**97%**	**97.23%**	**97.12%**	**97%**
Multiple Accuracy	**95%**	**95.15%**	**95.13%**	**95.08%**
Recall	**94.4%**	**94.1%**	**90%**	**90%**
Precision	**93.3%**	**93.3%**	**92.2%**	**92.1%**
f1-score	**93.24%**	**93.1%**	**90%**	**90.2%**

Towards the end of this paper, we will compare the data with the literature in the academic community that has used this open-source dataset. The focus will be on binary and multi-class accuracy, as well as other machine learning performance metrics.

Based on Table V, the bold text represents our research method. The other five references also use the ORNL dataset, with around 5,000 samples, indicating they averaged results from 15 datasets.

The binary accuracy of the other references is based on simulating the ORNL binary labeled dataset directly. In contrast, we use a stacking model with an auxiliary classifier for initial classification of natural and attack events.

TABLE VI
THE COMPARISON OF PERFORMANCE WITH ORNL REFERENCE

Research Method	Adaboost + JRipper	Common-Path Mining	Random Forest	RBM	GBFS
Reference	[6]	[7]	[8]	[9]	[10]
Data Instance	5,000	5,000	78,307	5,000	5,000

Train-Test Split	9:1		8:2	7:3	
Binary Accuracy	95%	95%	97.12%	97%	97%
Multiple Accuracy	89%	93%	95.13%	94%	92%
Recall	75%		90%	92%	92.5%
Precision	80%		92.2%	93%	92.4%
f1-score	80%		90%	93%	92.44%

References [6] and [8] in Table VI use basic machine learning models for intrusion detection. [6] does not consider feature reduction or preprocessing, while [8] employs feature engineering using feature correlation and information gain, resulting in a larger sample size.

Reference [7] reduces multi-class features using data formatting based on PMU measurement values, resulting in 7 features. We use Min-Max Scaler for quantizing PMU values into a range of 0 to 1, aiding the stacking model in precise event classification.

In summary, while the literature in Table V and Table VI uses the same dataset, each method varies, leading to different sample sizes. However, based on multi-class accuracy and other metrics, our intrusion detection model for power systems remains highly competitive.

VI. Conclusion

This paper proposes an ensemble learning method based on stacking for intrusion detection in power systems. The proposed model architecture can simultaneously determine the initial classification of events as either natural or attack events, further categorize them into specific attack types, and even identify which location in the power lines the fault occurred.

The preprocessing method involves encoding the dataset and adding binary labels for auxiliary classifiers conducive to the secondary learners in the stacking model, which is validated further in the experimental results.

The accuracy of the auxiliary classifier is found to be 97% for the averaged test data of the 15 datasets from the ORNL dataset, while the accuracy of the output layer for detecting attack events is 95%. Performance metrics for evaluating IDS, such as Recall, Precision, and F1-score, are reported at 94%, 93%, and 93% respectively. The experimental results also utilize a confusion matrix to provide a more intuitive understanding of the classification detection rates for each attack event.

Compared to other literature using the ORNL dataset for IDS, this approach demonstrates higher detection performance, particularly with the multi-class output layer achieving higher accuracy than most literature.

Reference

[1] A.V. Jha, B. Appasani, A.N. Ghazali, P. Pattanayak, D.S. Gurjar, E. Kabalci, and D. K. Mohanta, “Smart grid cyber-physical system: communication technologies, standards and challenges,” *Wireless Network.*, vol. 27, no, 4, pp. 2595–2613, May 2021, doi:10.1007/s11276-021-02579-1.

[2] C. Aggarwal, “*Data Classification: Algorithms and Applications*,” USA: New York, CRC Press, 2015, pp. 498-500

[3] Mississippi State University Critical Infrastructure Protection Center.(Apr. 2014). Industrial Control System Cyber Attack Data Set.[Online].Available: http://www.ece.msstate.edu/wiki/index.php/ICS_ Attack_Dataset

[4] F. Sabahi, and A. Movaghar “Intrusion Detection: A Survey,” in *Proceedings of the 2008 Third International Conference on Systems and Networks Communications*, Sliema, Malta, 26-31 October 2008, pp. 23-26, doi:10.1109/ICSNC.2008.44.

[5] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Luric, and A. Dosovitskiy, “Mlp-mixer: An all-mlp architecture for vision,” *Neural Information Processing Systems* ,2021, arXiv:2105.01601, 2021.

[6] R. C. Borges Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," *International Symposium on Resilient Control Systems (ISRCS)*, Denver, CO, USA, 2014, pp. 1-8, doi: 10.1109/ISRCS.2014.6900095.

[7] S. Pan, T. Morris, and U. Adhikari, ‘‘Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems,’’ I*EEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 3104–3113, Nov. 2015, doi: 10.1109/TSG.2015.2409775.

[8] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu and I. Khalil, "An Integrated Framework for Privacy-Preserving Based Anomaly Detection for Cyber-Physical Systems," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 1, pp. 66-79, 1 Jan.-March 2021, doi: 10.1109/TSUSC.2019.2906657.

[9] S. Y. Diaba, M. Shafie-Khah and M. Elmusrati, "Cyber Security in Power Systems Using Meta-Heuristic and Deep Learning Algorithms," *IEEE Access*,. vol. 11, pp. 18660-18672, Feb 2023, doi:10.1109/ACCESS.2023.3247193

[10] Upadhyay, J. Manero, M. Zaman and S. Sampalli, "Gradient Boosting Feature Selection With Machine Learning Classifiers for Intrusion Detection on Power Grids," in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 1104-1116, March 2021, doi: 10.1109/TNSM.2020.3032618