

Improving intrusion detection in SCADA systems using stacking ensemble of tree-based models

Duc-Duong Nguyen^{1,2}, Minh-Thuy Le¹, Thanh-Long Cung¹

¹School of Electrical Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

²Faculty of Electricity, University of Economics-Technology for Industries, Hanoi, Vietnam

Article Info

Article history:

Received Aug 6, 2021

Revised Nov 6, 2021

Accepted Jan 10, 2022

Keywords:

eXtreme gradient boosting

Intrusion detection

Light gradient boosting machine

Multilayer perceptron

Random forest

SCADA system

ABSTRACT

This paper introduces a stacking ensemble model, which combines three single models, to improve intrusion detection in supervisory control and data acquisition (SCADA) systems. The first layer of the proposed model is the combination of random forest, light boosting gradient machine, and eXtreme gradient boosting models. We use an multilayer perceptron (MLP) network as a meta-classifier of the model. The proposed model is optimized and tested on an international dataset (gas pipeline dataset). The tested results show an accuracy of 99.72% with the f1-score of 99.72% for binary classification tasks (attacked or non-attacked detection). For categorical tasks, the detection rates of almost all attack types are higher than 97.55% (except for denial of service (DoS)-95.17%), with an overall accuracy of 99.62%.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Thanh-Long Cung

School of Electrical Engineering, Hanoi University of Science and Technology

Dai-Co-Viet street, Hai-Ba-Trung district, Hanoi, Vietnam

Email: long.cungthanh@hust.edu.vn

1. INTRODUCTION

Nowadays, supervisory control and data acquisition (SCADA) systems are widely used in power transformer stations and industrial factories. To keep the regular operation of such systems, data collection, data processing, and, ensuring the integrity of the data are very important. SCADA systems can be attacked not only on the physical infrastructures but also on the communication and supervisory control layers, as shown in Figure 1 [1]. A1, A2, A3 are attack points aimed at the supervisory control layer and through applications on a web server to spread viruses that destroy the control and supervising network configuration. The attack at point A4 is to occupy access to communication channels between the control center and stations. A5, A6 are attack points aimed at the communication link between MTU and PLC/RTU. A7 is an attack point on the network connection between factories and their contractors. The attack at point A8 aims at field terminal devices. The attack at points A9 and A10 aim at the signal lines from controllers to actuators, and the feedback signals from sensors to controllers, respectively. At point A0, attacks are all direct mechanical impacts to physical layer devices of SCADA systems. To exploit SCADA protocol weaknesses, attackers usually use four general types of attack: interception, interruption, modification, and fabrication [2], [3]. They can target the network infrastructure, RTU/PLC, and HMI of SCADA systems. Therefore, data safety studies for industrial control systems are of great interest. There are two main directions: the research on new attack types to test the ability of information security methods, and the second research direction is to focus on building methods to detect intrusions.

According to the first research direction, it is possible to classify several attack methods as denial of service (DoS) attacks, data integrity attacks (between layers, or in each layer of a control system) such as

falsifying information, inserting fake information [5]. The second research direction, which is on information security, is currently receiving much attention [6]. For data intrusion detection problem, traditional machine learning approaches [7]-[11] and deep learning neural network architectures (for big data problems) [12]-[16] are widely used. Otherwise, many attempts to build datasets for SCADA intrusion detection have been accomplished [14], [17]-[19].

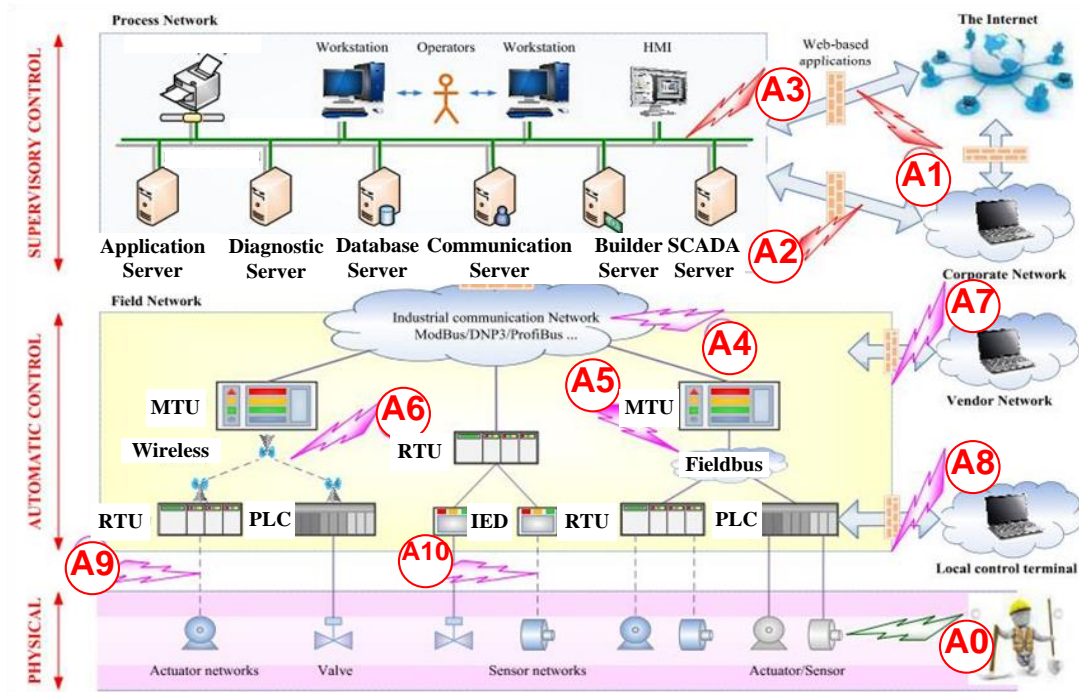


Figure 1. Possible attack points to SCADA systems [4]

A standard simulation dataset (gas pipeline dataset) of data falsification attacks in industrial networks is provided by Mississippi State University's in-house SCADA lab in 2015 [17]. Using this dataset, many methods are proposed to improve the ability of intrusion detection. The authors presented an approach of using the random forest (RF) model to examine three data-split strategies for classifying categories of SCADA attacks [20]. 70-30 split strategy results showed 5-17% and 2-8% improvement of accuracy for the classification of reading response attacks and write command attacks, respectively. The authors proposed a hybrid model which consists of a GoogLeNet neural network and a long short-term memory neural network (GoogLeNet-LSTM) for intrusion detection of industrial control systems [21]. The accuracy of the proposed model reached 97.56%. Two separate models (support vector machine (SVM) and RF) were trained and evaluated [3]. The proposed RF model reached the accuracies of 99.58% and 99.41% for binary and categorical classification tasks, respectively. An ensemble of stacked autoencoder model proposed in [22] achieved the accuracy of 95.86% and 93.83% for f1-score. The authors of [23] presented a LSTM model which showed the accuracy and f1-score of 92% and 85% respectively. Different classification methods (including K-means, Naïve Bayes (NB), principal component analysis-singular value decomposition (PCA-SVD), gaussian mixture model (GMM)) were examined for detecting intrusion of gas pipeline dataset [24]. Among them, the K-means model showed the best accuracy of 83.19%. Some machine learning models (HoeffdingTree, NaiveBayes, RandomTree, BayesNet and OneR) were investigated with some techniques of cost-sensitive learning and Fisher's (linear) discriminant analysis (FDA) [25]. The random tree algorithm combined with cost-sensitive learning enhancements showed the best prediction performance with the accuracy of 97.8%.

This paper proposes a stacking ensemble model that combines three single models to detect data intrusion in SCADA systems. The gas pipeline dataset will be used to validate our proposed model. The following parts of the paper are organized as shown in: the second section presents the dataset and methodology; experiments and results are shown in the third section; finally, some conclusions and our future works are presented in section 4.

2. DATA SET AND METHODOLOGY

2.1. Gas pipeline dataset

The gas pipeline dataset consists of 274,628 instances, and each instance contains 17 features. These instances indicate the state and parameters of Modbus frames in a gas pipeline SCADA system, with three different types of labels showing the state of the network. The 17 features of each instance indicate the network (Address, CRC, C/R, ...) and payload information in Table 1. The network status information is divided into three groups: binary results, categorical results, and specific results. In this work, only binary and categorical results are used. The Binary results contain two states: Normal and Attacked states. The categorical results consist of 7 types of attacked and normal states in Table 2. The dataset was also introduced as a heavily imbalanced dataset, with the number of Normal instances accounts for 78.1% and the number of attacked samples comprises 21.9% of the total cases.

Table 1. Three first rows of the raw dataset with 17 features and 2 label types

Address	Function	Length	Payload	CRC	C/R	Timestamp	Binary result	Categorical result
4	3	16	?,?,?,?,?,?,?,?,??	12869	1	1418682163.170388	0	0
4	3	46	?,?,?,?,?,?,?,?,?,0.689655	12356	0	1418682163.269946	0	0
4	16	90	10,115,0.2,0.5,1,0,0,1,0,0,?	17219	1	1418682164.995590	0	0

Table 2. Description, category of the attacks

Attack description	Threat type	Abbreviation
Normal	N/A	Normal (0)
Naïve malicious response injection	Modification/Fabrication	NMRI (1)
Complex malicious response injection	Modification/Fabrication	CMRI (2)
Malicious state command injection	Modification/Fabrication	MSCI (3)
Malicious parameter command injection	Modification/Fabrication	MPCI (4)
Malicious function code injection	Modification/Fabrication	MFCI (5)
DoS	Interruption	DoS (6)
Reconnaissance	Interception	Recon (7)

2.2. Data pre-processing

As seen in Table 1, many payload features are not available, making it impossible to train any classifier on these data since machine learning models usually require fixed-size inputs. Missing data commonly occurs in machine learning, and many approaches can be used to handle this problem. In this research, we used the "keep prior values" strategy, which was one of four methods demonstrated in [3] to impute all missing values of the dataset. In this way, all missing values of a row in the dataset will be attributed to the nearest non-missing row values above/below it (Table 3). After handling missing data, the dataset is divided into a training and a testing set (with the rate of 80%-20%, respectively) to train and validate our classification method, both training and testing set have the same attack/normal ratio. The min-max normalization is applied to normalize datasets.

Table 3. Three first rows of the raw dataset after using the "keep prior values" imputation strategy

Address	Function	Length	Payload	CRC	C/R	Timestamp	Binary result	Categorical result
4	3	16	10,115,0.2,0.5,1,0,0,1,0,0,0.689655	12869	1	1418682163.170388	0	0
4	3	46	10,115,0.2,0.5,1,0,0,1,0,0,0.689655	12356	0	1418682163.269946	0	0
4	16	90	10,115,0.2,0.5,1,0,0,1,0,0,0.689655	17219	1	1418682164.995590	0	0

2.3. Machine learning techniques

2.3.1. Classification metrics

The most ubiquitous metric for classification tasks is accuracy, which can be formulated as the ratio of total correct predictions to all predictions. As described earlier, the gas pipeline dataset is exceptionally imbalanced, making the accuracy be an ineffective metric for classification tasks. For example, regarding this dataset, if all instances are predicted as 0 (normal state), the accuracy, in this case, is 78.1% (equal to the proportion of the normal state), which is considerably high for a classification task. Still, the model's performance is awful. It is even worse when applying to intrusion detection, where incorrectly detecting any attack to be normal is more severe than misclassifying a normal state as an attack. Because of that, using other metrics to evaluate a classification model rather than the accuracy only is necessary. Notions and

formulas of some standard metrics-which are prevalent in imbalanced dataset classification tasks-are presented in (1) to (4). The accuracy score is defined as in (1):

$$accuracy = \frac{TP+TN}{TP+FP+FN+TN} * 100\% \quad (1)$$

Precision is defined as the ratio of correctly predicted positive observations to the total predicted positive observations, which as shown in (2):

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

Recall is defined as the ratio of correctly predicted positive observations to all observations in class 0 (Normal). It is indicated as in (3):

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

f1-score is the weighted average of precision and recall, as shown in (4):

$$f1 - score = 2 * \frac{recall * precision}{recall + precision} \quad (4)$$

Where true positives (TP): the number of instances in class 0 (normal) which the model indeed predicts; true negatives (TN): the number of instances in class 1 (attack) which the model truly predicts; false positives (FP): the number of instances in class 0 (normal) which are falsely predicted by the model. false negatives (FN): the number of instances in class 1 (attack) which the model falsely predicts.

The values of precision, recall, and *f1-score* are non-negative numbers and smaller than one. High precision relates to the high rate of correctly classifying positive instances. A high value of recall means that the true positive rate is high (the rate of misclassifying positive instances is low). The *f1-score* takes both false positives and false negatives into account, and it is used to seek a balance between precision and recall, which is significant in evaluating an imbalanced dataset.

2.3.2. Classification models

In this work, we choose four different models, which are RF [26], light gradient boosting machine (LGBM) [27], eXtreme gradient boosting (XGBoost) [28], and multilayer perceptron (MLP), to construct our ensemble model for the intrusion detection task. RF, LGBM, and XGBoost are the same type of tree-based models. These models are chosen because of their fast training-speed and effectiveness in classification tasks. MLP will be used to make the final decision.

In machine learning, there are various types of models that can be used for classification tasks. Every model has its own merits and defects. This type of model can perform well in a specific situation but maybe bad for others. Therefore, to combine the advantages of individual models, many approaches called ensemble learnings were developed. Ensemble learning combines multiple models to build a stronger one to solve a particular problem. Some popular ensemble algorithms are boosting, bagging, and stacking. Bagging (stands for bootstrap aggregating) is a way to decrease the variance of the prediction. This is done by generating additional data for training from the original dataset. By increasing the size of the training set, the variance of predictions can be decreased to increase the reliability of predictions. The boosting algorithm first uses subsets of the original data to produce a series of averagely performing models and then "boosts" their performance by combining them, using a particular cost function (majority vote). Unlike bagging, in classical boosting, the subset creation is not random. It depends on the performance of previous models. Every new subset contains the elements that previous models misclassified. For the stacking approach, first, several models are applied to original data. Then, a meta-level classifier is used to make final decisions. This classifier uses outputs of every first-level model as its input data. For RF, LGBM, and XGBoost, RF is a bagging model, while LGBM and XGBoost are boosting models. These three models use multiple decision tree classifiers to generate the final prediction.

In this work, we propose an ensemble model, which uses a stacking strategy to combine three different classifiers. The structure of our stacking model is presented in Figure 2. The best hyper-parameter set of each model will be chosen in the first level, using cross-validation (CV) and random search. When all hyper-parameters for each first-level model are selected, these classifiers will be trained on the whole training dataset. Outputs of these models, then, will be used to train a multilayer perceptron neural network (MLP) as the meta-classifier of the stacking model.

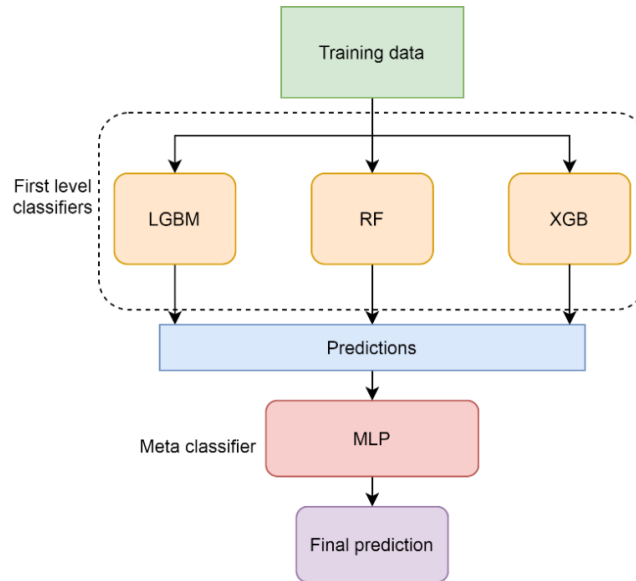


Figure 2. The architecture of the stacking model

3. IMPLEMENTATION AND RESULTS

In this work, to implement the RF model, the Scikit-learn package was used [29]. The LightGBM library was used for the LGBM model, and the XGBoost package was used for the XGB model. The Scikit-learn package was also used to normalize data, train and evaluate models. Finally, the stacking model was trained by using the Mlxtend package. All our codes are written in Python language.

3.1. Model parameter selection

The parameter selection for first-level models is implemented on training data, using a 5-fold CV and random search. By this way, the training data was equally divided into five separate subsets. Each first-level model (with a hyperparameter set) will be trained five times, separately. At each time, four subsets will be used for training and the remaining one will be used for evaluation. Then, the average accuracy of five folds will be used to choose the best hyperparameter set of each first-level model. The results are shown in Figure 3. Twenty tree-based classifiers with different parameter sets are tested and evaluated.

According to the binary detection task (2 classes of Normal/Attacked), as shown in Figure 3(a), the LGBM model reaches the best accuracy of 95.08%, with the hyper-parameter set of {'n_estimators': 104, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'auto', 'max_depth': 61, 'bootstrap': False}. For RF model, the best accuracy score is 96.56%, with the hyper-parameter set of {'colsample_bytree': 0.4029038429583326, 'max_depth': 39, 'min_child_samples': 293, 'min_child_weight': 1e-05, 'n_estimators': 865, 'num_leaves': 48, 'scale_pos_weight': 1, 'subsample': 0.20256029767506548}. For XGB model, the highest accuracy is 98.51%, achieved with the parameter of {'colsample_bytree': 0.6173735153519851, 'gamma': 1.5, 'max_depth': 94, 'min_child_weight': 5, 'n_estimators': 163, 'subsample': 0.935523447634425}.

Regarding the categorical detection task (8 classes: 1 Normal/7 types of Attacked), Figure 3(b) shows that, the LGBM model reaches an accuracy of 96.68% with the hyper-parameter set of {'n_estimators': 272, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'auto', 'max_depth': None, 'bootstrap': True}. The best accuracy score of RF model is 97.99%, achieved with the hyper-parameter set of {'colsample_bytree': 0.9182559913420497, 'min_child_samples': 336, 'min_child_weight': 0.01, 'n_estimators': 824, 'num_leaves': 47, 'reg_alpha': 0.1, 'reg_lambda': 0, 'scale_pos_weight': 2, 'subsample': 0.46476797298571926}. For XGB model, the highest accuracy is 99.21%, achieved with the parameter set of {'colsample_bytree': 0.9048840146153558, 'gamma': 0.5, 'max_depth': 38, 'min_child_weight': 10, 'n_estimators': 242, 'subsample': 0.600878069464399}.

The models with the highest accuracy score were chosen, trained, and evaluated on a full training/testing dataset. The final efficiency of each model is given in Table 4 and Table 5. All predictions of these models will be used to optimize hyper-parameters of the MLP meta-classifier. For MLP, we fix the number of hidden layers as one. Then, the number of neurons in the hidden layer is optimized using a random search strategy. The optimization results are shown in Figure 4. For the binary detection task, the stacking model reaches an accuracy of 99.72%, with 24 neurons in the hidden layer of the MLP meta-classifier in Figure 4(a). For the categorical detection task, the accuracy of the stacking model is up to 99.62%, with 97 neurons in the hidden layer of the MLP in Figure 4(b).

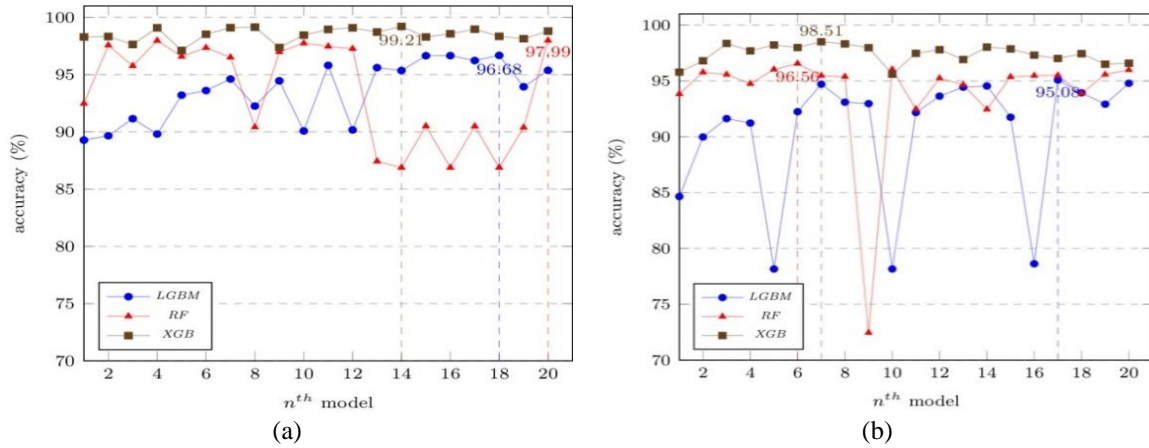


Figure 3. Performance of first-level models by random search CV (a) for 2 classes and (b) 8 classes

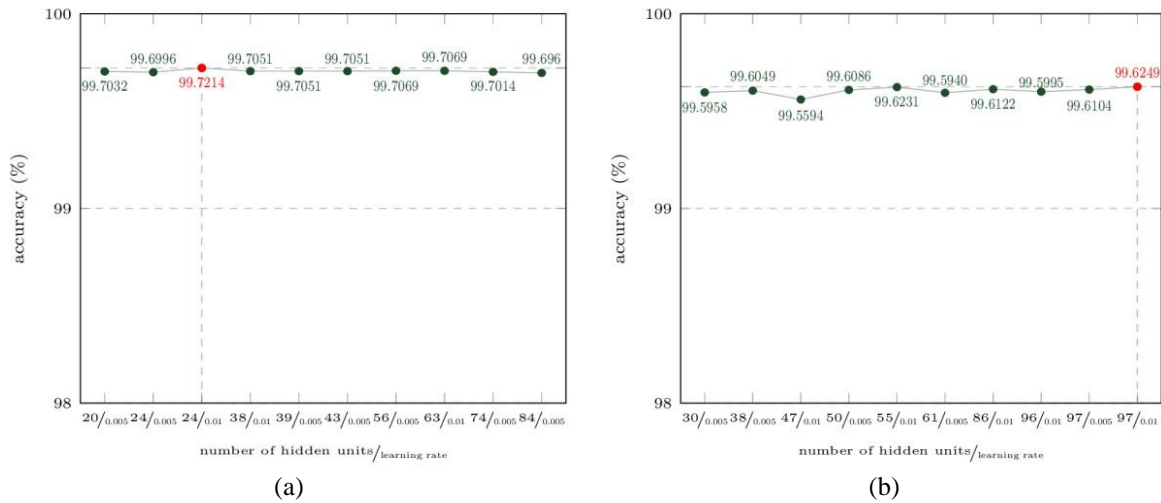


Figure 4. Performance of meta-classifier on the test set by the random search (a) for 2 classes and (b) 8 classes

Table 4. Performance of each classifier on test set for the binary detection task

Model	classification report-2 classes (7 attack types + normal)									
	Precision	LGBM		RF			XGB		Support	
		Recall	f1-score	precision	Recall	f1-score	precision	Recall	f1-score	Support
Normal(0)	98.76037	95.24922	96.97302	99.46873	98.90639	99.18677	99.93243	99.46657	99.69895	43117
Attack(1)	82.398	94.89835	88.20751	96.06994	98.06221	97.05585	98.08493	99.75442	98.91263	11809
Weighted avg	95.1725	95.18261	95.05638	98.72181	98.72556	98.72082	0.995295	0.995285	0.99527	54926
Accuracy	95.18261			98.72556			99.52846			

Table 5. Performance of each classifier on test set for 8-classes detection task

Model	classification report - 8 classes (7 attack types + normal)										
	Precision	LGBM			RF			XGB			Support
		Recall	f1-score	Precision	Recall	r1-score	Precision	Recall	f1-score	Support	
Normal (0)	98.80697	97.75462	98.27798	99.53863	98.59669	99.06542	99.93709	99.44814	99.69201	43127	
NMRI (1)	76.66022	82.17001	79.31955	86.8472	93.02486	89.82994	95.68021	97.63158	96.64604	1520	
CMRI (2)	80.13042	85.89638	82.91328	88.4158	94.27403	91.25099	96.39432	98.58768	97.47867	2549	
MSCI (3)	91.64557	96.92102	94.2095	98.29114	99.16986	98.72854	97.8481	99.93536	98.88072	1547	
MPCI (4)	99.43655	100	99.71748	97.15826	99.89924	98.50969	98.67712	99.97518	99.32191	4029	
MFCI (5)	100	98.89001	99.44191	100	98.79032	99.39148	100	99.08999	99.54292	989	
DoS (6)	79.77011	93.531	86.10422	94.25287	98.79518	96.47059	93.7931	99.7555	96.68246	409	
Recon (7)	97.29032	100	98.62655	97.16129	98.56021	97.85575	97.54839	100	98.75898	756	
Weighted avg	97.12042	96.98503	97.03749	98.43663	98.37236	98.39378	99.42612	99.41376	99.41707	54926	
Accuracy	96.98503			98.37236			99.41376				

3.2. Prediction results

All classification metrics (precision, recall, f1-score) of three individual models (LGBM, RF, XGB) for 2-class and 8-class detection tasks are shown in Table 4 and Table 5, respectively. As seen in both tables, the XGB is the best among three individual classifiers, and the LGBM is the worst model. The f1-score of the XGB model reaches 99.53% for the binary detection task and 98.37% for the categorical detection task.

Table 6 and Table 7 show the prediction results of our proposed stacking models. The results of both stacking models are more accurate than that of three individual classifiers. For the binary task, the accuracy and f1-score of the stacking model are the same, with 99.72%. For categorical tasks, the accuracy and f1-score are 99.62% and 99.63%, respectively. Moreover, Figure 5 shows the detailed quality of two stacking models through confusion matrices. The detection rates (recall) of attacked and normal states are greater than 99.32%, with an overall accuracy of 99.83% in Figure 5(a). For categorical tasks, the detection rates of almost all attack types are higher than 97.55% (except for DoS-95.17%), with an overall accuracy of 99.62% in Figure 5(b).

Table 6. Performance of the stacking model on test set for 2-class detection task

2 classes	Precision	Recall	f1-score	Support
Normal (0)	99.80898	99.83456	99.82177	42916
Attack (1)	99.40828	99.31724	99.36274	12010
Weighted avg	99.72136	99.72144	99.7214	54926
Accuracy	99.72144			

Table 7. Performance of the stacking model on test set for 8-class detection task

8 classes	Precision	Recall	f1-score	Support
Normal (0)	99.86019	99.75095	99.80554	42963
NMRI (1)	97.54997	97.48711	97.51853	1552
CMRI (2)	98.04373	98.57308	98.30769	2593
MSCI (3)	99.36709	99.43002	99.39854	1579
MPCI (4)	99.75502	99.85287	99.80392	4078
MFCI (5)	100	99.39148	99.69481	986
DoS (6)	95.17241	99.51923	97.2973	416
Recon (7)	97.93548	100	98.95698	759
Weighted avg	99.62759	99.62495	99.62568	54926
Accuracy	99.62495			

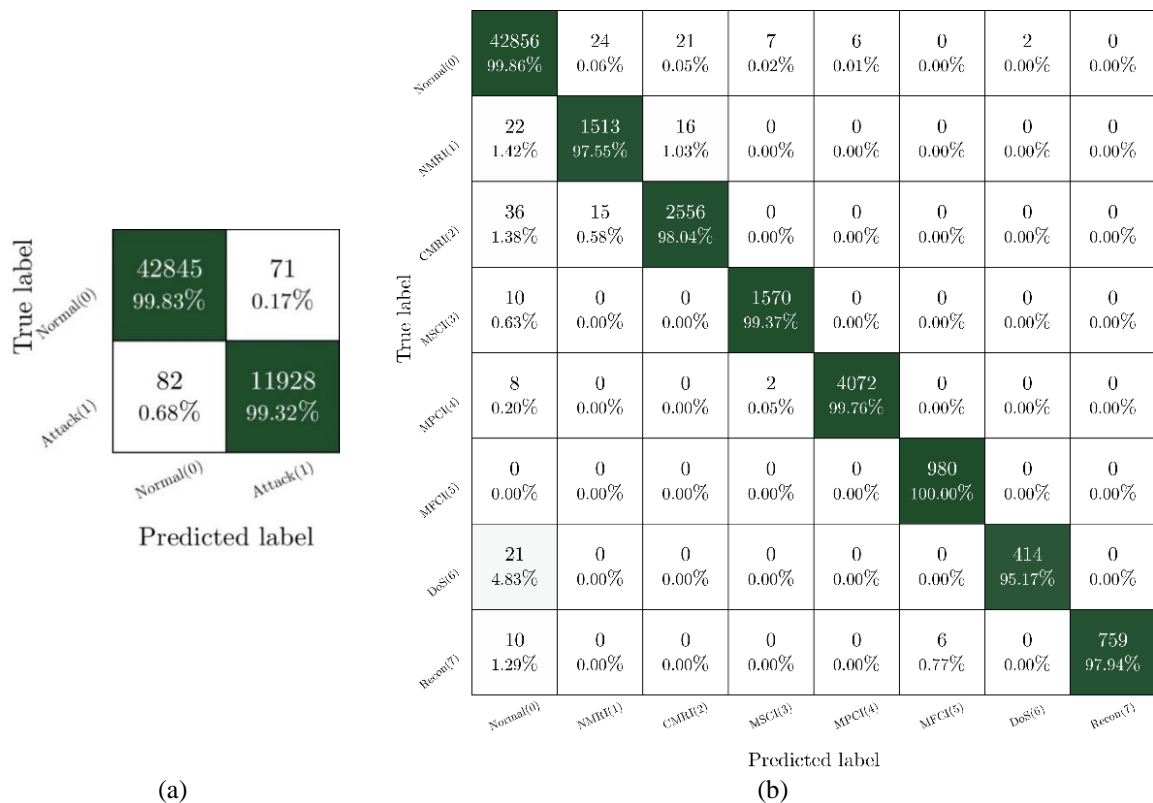


Figure 5. Detailed quality through confusion matrices (a) confusion matrix of binary classification task and (b) categorical classification task

A comparison of prediction results of our proposed model with other recent works which implemented on the same gas pipeline dataset is given in Table 8. As seen in Table 8, almost works take single classification, except the ensemble of SAE model in [22], which achieved an accuracy of 95.86% and 93.83% for f1-score. Our proposed model showed a relatively high prediction rate for both binary and categorical tasks, as compare to another.

Table 8. Comparative results of our proposed model with other recent works

Methods	Binary task		Categorical task	
	Accuracy (%)	f1-score (%)	Accuracy (%)	f1-score (%)
Bagged tree [20]	98.2			
LSTM [23]	92	85	-	-
K-means, NB, PCA-SVD, GMM [24]	83.19 (K-means))	86.05 (NB)	-	-
Ensemble of SAE [22]	95.86	93.83	-	-
GoogLeNet-LSTM [21]	97.56	-	-	-
RF [22]	99.58	99.58	99.41	99.41
RandomTree [25]	97.8	-	-	-
Our ensemble model	99.72%	99.72	99.62	99.62

4. CONCLUSION

In this work, we have proposed one type of stacking model to improve the quality of intrusion detection in SCADA systems. The first layer of the proposed model is the combination of random forest, light boosting gradient machine, and eXtreme gradient boosting models. We use an MLP network as a meta-classifier of the model. The proposed model is optimized and tested on an international dataset (gas pipeline dataset). Testing results are a prospect, in which the detection accuracy is 99.72% and 99.62% for binary and categorical detection tasks, respectively. In our future works, all the binary, categorical and specific results in the gas pipeline dataset are considered. A variant version of the proposed stacking model will be developed and tested to deal with this problem.





REFERENCES

- [1] M. Lehto and P. Neittaanmäki, "Cyber security: Analytics, technology and automation," Springer, vol. 78, 2015, doi: 10.1007/978-3-319-18302-2.
- [2] S. East, J. Butts, M. Papa, and S. Sheno, "A Taxonomy of Attacks on the DNP3 Protocol," in *International Conference on Critical Infrastructure Protection*, 2009, pp. 67-81, doi: 10.1007/978-3-642-04798-5_5.
- [3] R. L. Perez, F. Adamsky, R. Soua and T. Engel, "Machine Learning for Reliable Network Attack Detection in SCADA Systems," *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 633-638, doi: 10.1109/TrustCom/BigDataSE.2018.00094.
- [4] V. L. Do, "Sequential detection and isolation of cyber-physical attacks on SCADA systems," Ph.D. dissertation, University of Technology of Troyes (UTT), 2015.
- [5] A. Hijazi, A. El Safadi, and J.-M. Flaus, "A Deep Learning Approach for Intrusion Detection System in Industry Network.," in *Lebanese University 5 décembre, 12:50 The first international conference on Big Data and Cybersecurity intelligence (BDCS Intell' 2018)*, Beirut, Lebanon, 2018, pp. 55-62.
- [6] I. Butun, S. D. Morgera and R. Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 266-282, First Quarter 2014, doi: 10.1109/SURV.2013.050113.00191.
- [7] J. M. Beaver, R. C. Borges-Hink and M. A. Buckner, "An Evaluation of Machine Learning Methods to Detect Malicious SCADA Communications," *2013 12th International Conference on Machine Learning and Applications*, 2013, pp. 54-59, doi: 10.1109/ICMLA.2013.105.
- [8] B. M. Susanto, "Naïve Bayes Decision Tree Hybrid Approach for Intrusion Detection System," *Bulletin of Electrical Engineering and Informatics*, vol. 2, no. 3, pp. 225-232, 2013, doi: 10.12928/eei.v2i3.208.
- [9] L. A. Maglaras and J. Jiang, "Intrusion detection in SCADA systems using machine learning techniques," *2014 Science and Information Conference*, 2014, pp. 626-631, doi: 10.1109/SAL.2014.6918252.
- [10] J. Gao *et al.*, "LSTM for SCADA Intrusion Detection," *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 2019, pp. 1-5, doi: 10.1109/PACRIM47961.2019.8985116.
- [11] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain and A. Nawaz, "HML-IDS: A Hybrid-Multilevel Anomaly Prediction Approach for Intrusion Detection in SCADA Systems," in *IEEE Access*, vol. 7, pp. 89507-89521, 2019, doi: 10.1109/ACCESS.2019.2925838.
- [12] H. Yang, L. Cheng and M. C. Chuah, "Deep-Learning-Based Network Intrusion Detection for SCADA Systems," *2019 IEEE Conference on Communications and Network Security (CNS)*, 2019, pp. 1-7, doi: 10.1109/CNS.2019.8802785.
- [13] J. Majidpour and H. Hasanzadeh, "Application of deep learning to enhance the accuracy of intrusion detection in modern computer networks," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, p. 1724, 2020, doi: 10.11591/eei.v9i3.1724.
- [14] T. Morris and W. Gao, "Industrial control system traffic data sets for intrusion detection research," in *International Conference on Critical Infrastructure Protection*, 2014, pp. 65-78, doi: 10.1007/978-3-662-45355-1_5.
- [15] J. Gao *et al.*, "Omni SCADA Intrusion Detection Using Deep Learning Algorithms," in *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 951-961, 15 Jan.15, 2021, doi: 10.1109/JIOT.2020.3009180.
- [16] A. N. Sokolov, S. K. Alabugin and I. A. Pyatnitsky, "Traffic Modeling by Recurrent Neural Networks for Intrusion Detection in Industrial Control Systems," *2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, 2019, pp. 1-5, doi: 10.1109/ICIEAM.2019.8742961.





- [17] I. P. Turnipseed, "A new scada dataset for intrusion detection research," Mississippi State University, 2015.
- [18] A. Lemay and J. M. Fernandez, "Providing SCADAS network data sets for intrusion detection research," in *9th Workshop on Cyber Security Experimentation and Test (CSET)*, 2016, doi: 10.5555/3241067.3241073.
- [19] M. Du, S. Ma and Q. He, "A SCADA data based anomaly detection method for wind turbines," *2016 China International Conference on Electricity Distribution (CICED)*, 2016, pp. 1-6, doi: 10.1109/CICED.2016.7576060.
- [20] K. M. Paramkusem and R. S. Aygun, "Classifying categories of SCADA attacks in a big data framework," *Annals of Data Science*, vol. 5, no. 3, pp. 359-386, 2018, doi: 10.1007/s40745-018-0141-8.
- [21] A. Chu, Y. Lai, and J. Liu, "Industrial control intrusion detection approach based on multiclassification GoogLeNet-LSTM model," *Security and Communication Networks*, vol. 2019, 2019, doi: 10.1155/2019/6757685.
- [22] A. Al-Abassi, H. Karimipour, A. Dehghantanha and R. M. Parizi, "An Ensemble Deep Learning-Based Cyber-Attack Detection in Industrial Control System," in *IEEE Access*, vol. 8, pp. 83965-83973, 2020, doi: 10.1109/ACCESS.2020.2992249.
- [23] C. Feng, T. Li and D. Chana, "Multi-level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM Networks," *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2017, pp. 261-272, doi: 10.1109/DSN.2017.34.
- [24] S. N. Shirazi *et al.*, "Evaluation of Anomaly Detection techniques for SCADA communication resilience," *2016 Resilience Week (RWS)*, 2016, pp. 140-145, doi: 10.1109/RWEEK.2016.7573322.
- [25] A. Choubineh, D. A. Wood, and Z. Choubineh, "Applying separately cost-sensitive learning and Fisher's discriminant analysis to address the class imbalance problem: A case study involving a virtual gas pipeline SCADA system," *International Journal of Critical Infrastructure Protection*, vol. 29, p. 100357, 2020, doi: 10.1016/j.ijcip.2020.100357.
- [26] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News*, vol. 2, no. 3, pp. 18-22, 2002.
- [27] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3149-3157, doi: 10.5555/3294996.3295074.
- [28] T. Chen and C. Guestrin, "{XGBoost}: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794. doi: 10.1145/2939672.2939785.
- [29] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825-2830, 2011, doi: 10.5555/1953048.2078195.

BIOGRAPHIES OF AUTHORS







Duc Duong Nguyen     received his B. Eng (2009) and M. Sc (2011) degree in Control and Automation Engineering at Hanoi University of Science and Technology. From 2009 to 2012, he worked as a researcher at Hitech center - Hanoi University of Science and Technology. Since 2013, he has been working as a lecturer at the Faculty of Electricity, University of Economics - Technology for Industries. He is being a PhD student in Control and Automation Engineering, at Hanoi University of Science and Technology. His main research areas are intrusion detection in SCADA systems, industrial information systems, process control, and automation of production process. He can be contacted at ndduong86.ddt@uneti.edu.vn



Minh Thuy Le     received her engineer (2006), M.S (2008) degree in Electrical Engineering from Hanoi University of Science and Technology and PhD (2013) degree in Optics and Radio Frequency from Grenoble Institute of Technology, France. She is lecturer and also a Group leader of Radio Frequency group at Department of Instrumentation and Industrial Informatics (3I), School of Electrical Engineering (SEE), Hanoi University of Science and Technology (HUST). Her current interests include built-in antenna, antenna array, beamforming, metamaterials, indoor localization, RF energy harvesting, wireless power transfer and wireless sensor network. She can be contacted at: thuy.leminh@hust.edu.vn.



Thanh Long Cung     received his B. Eng degree in Measurement and Automatic Control, in 2000, and his M. Sc degree in Measurement and Control Systems, in 2002, at Hanoi University of Science and Technology, Vietnam. He received his PhD degree in Electronics-Electrotechnique-Automation, in 2012, at Ecole Normale Supérieure Paris-Saclay, France. He is currently a researcher/lecturer at the School of Electrical Engineering, Hanoi University of Science and Technology. His research interests include electromagnetic non-destructive testing/evaluation, human emotion recognition, sensors, and signal processing. He can be contacted at: long.cungthanh@hust.edu.vn.