big data and cognitive computing

Article

Exploring Ensemble-Based Class Imbalance Learners for Intrusion Detection in Industrial Control Networks

Maya Hilda Lestari Louk [1] and **Bayu Adhi Tama** [2,*]

[1] Department of Informatics Engineering, University of Surabaya, Surabaya 60293, Indonesia; mayalouk@staff.ubaya.ac.id
[2] Data Science Group, Institute for Basic Science (IBS), Daejeon 34141, Korea
* Correspondence: bayuat@ibs.re.kr

Abstract: Classifier ensembles have been utilized in the industrial cybersecurity sector for many years. However, their efficacy and reliability for intrusion detection systems remain questionable in current research, owing to the particularly imbalanced data issue. The purpose of this article is to address a gap in the literature by illustrating the benefits of ensemble-based models for identifying threats and attacks in a cyber-physical power grid. We provide a framework that compares nine cost-sensitive individual and ensemble models designed specifically for handling imbalanced data, including cost-sensitive C4.5, roughly balanced bagging, random oversampling bagging, random undersampling bagging, synthetic minority oversampling bagging, random undersampling boosting, synthetic minority oversampling boosting, AdaC2, and EasyEnsemble. Each ensemble's performance is tested against a range of benchmarked power system datasets utilizing balanced accuracy, Kappa statistics, and AUC metrics. Our findings demonstrate that EasyEnsemble outperformed significantly in comparison to its rivals across the board. Furthermore, undersampling and oversampling strategies were effective in a boosting-based ensemble but not in a bagging-based ensemble.

Keywords: imbalanced data; cost-sensitive learning; ensemble-based models; intrusion detection; industrial control networks

check for updates

Citation: Louk, M.H.L.; Tama, B.A. Exploring Ensemble-Based Class Imbalance Learners for Intrusion Detection in Industrial Control Networks. *Big Data Cogn. Comput.* **2021**, *5*, 72. https://doi.org/10.3390/bdcc5040072

Academic Editors: Mohamed Eldefrawy, Mahmoud Eldefrawy and Giancarlo Fortino

Received: 22 October 2021
Accepted: 2 December 2021
Published: 6 December 2021

1. Introduction

As of today, a large number of cyber-attack vectors have put many organizations' critical infrastructure in jeopardy. A successful attack could have serious consequences, such as revenue loss, operational halting, and the disclosure of sensitive information. Furthermore, the complex nature of the infrastructure may lead to vulnerabilities and other unexpected risks. As a result, security mitigation and protection techniques should be prioritized. A potential defense system, such as an intrusion detection and prevention system, is required because it provides an alert when a signature match is detected and actively blocks traffic. It is deployed to supplement firewalls and access control to filter out any malicious activities within the computer network.

Classifier ensembles combine numerous weak learners, for example, base classifiers, to form a more robust learner. Each base learner is programmed and decides concerning a certain target class, while a combiner helps in making estimations [1]. Any classification method, such as decision trees, lazy classifiers, neural networks, or other types of learners, may be used as the base learner. According to its diversity, an ensemble scheme may use a single classifier to create a homogenous ensemble. In contrast, others may employ a variety of learners to create a heterogeneous ensemble [2]. Kuncheva [3] defined ensemble learners on four different perspectives—mixture level, learning algorithm level, attribute level, and data level. Similarly, Rokach [2] argued that an ensemble approach could be characterized by five critical components, namely mixer, diversification, orchestration structure, orchestration size, and commonality.

Such ensemble learners have been used in intrusion detection systems (IDSs) for decades, leading to increased detection accuracy as compared to any individual learners that comprise the ensemble [4]. Creating IDS aims to provide a security system that includes preventative actions against network attack activities. As the primary backbone of an IDS, a machine learning algorithm must be developed and configured appropriately to provide optimum security protection [5]. Because an IDS does not just need performance enhancement, the efficiency of the classifier (e.g., reduced computational cost) is also beneficial in providing a real-time detection method.

Intrusion detection in critical infrastructure (e.g., industrial control networks) is a domain research problem, where choosing a seemingly and computationally efficient algorithm is very challenging. It is a problematic task as each intrusion dataset is unique in terms of network architecture and skewed cyber-attack distributions (e.g., imbalanced data). Recognizing the importance of the aforementioned issues, this article conducts a comparative analysis of various class imbalance ensemble-based intrusion detection models, allowing researchers in this field to understand better the available best-performing ensemble models developed specifically for dealing with imbalanced data. This study focuses on class imbalance individual, bagging, boosting, and hybrid ensemble-based learners applied to power system attack detection, an area that has actually gained insufficient consideration in the present studies. In this work, we examine nine implementations, including cost-sensitive decision tree (CC4.5) [6], roughly balanced bagging (RBBagging) [7], random oversampling bagging (ROSBagging), random undersampling bagging (RUSBagging), synthetic minority oversampling bagging (SMOTEBagging) [8], random undersampling boosting (RUSBoosting) [9], synthetic minority oversampling boosting (SMOTEBoosting) [10], AdaC2 [11], and EasyEnsemble [12].

The aims of the paper lie in two different axes:

- To evaluate the effectiveness of class imbalance learners over a range of imbalance ratios.
- To provide an informed choice in choosing best-performing methods for identifying attacks in terms of performance accuracy and computational complexity.

The rest of this article is divided into four parts. The state-of-the-art techniques for detecting intrusion into a cyber-physical system are summarized in Section 2. Section 3 has a concise summary of the datasets and techniques utilized. Section 4 examines and delves into the result of the experiment benchmark, while Section 5 brings the study to a conclusion.

2. Related Work

A review study usually attempts to identify and classify previously published studies to represent their findings visually. Such an approach is restricted to reviewing the literature on the most frequent techniques for detecting attacks. It is, however, regarded as an unreliable guideline for identifying the best-performing classifiers. For instance, Alimi et al. [13] provided a survey map of research addressing supervised machine learning approaches for intrusion detection and classification in SCADA networks. The study analyzes and contrasts various conventional machine learning methods without considering other essential factors such as class imbalance. By using five publicly available standard intrusion datasets, Shahraki et al. [14] compared three different Adaboost ensembles, i.e., Real Adaboost, Gentle Adaboost, and Modest Adaboost. The study has several limitations despite providing an informed decision about using a boosting-based ensemble for intrusion detection. The performance benchmark is restricted to a small number of ensemble models and the absence of a statistical significance test. Modest Adaboost, on the other hand, has a greater prediction error than the rest algorithms.

Next, we present the most recent solutions for identifying threats and malicious activities in industrial control networks using ensemble techniques. The work of Anton et al. [15] compared SVM and random forest, finding that random forest outperformed SVM marginally. Khan et al. [16] used SMOTE algorithm to tackle the issue of class imbalance. At the same time, a two-level classifier is considered for classifying network anomalies in a gas pipeline

SCADA system. More recently, Upadhyay et al. [17] coupled recursive feature elimination using XGBoost and majority rule aggregation approaches for intrusion detection in a SCADA-based power grid. A total of nine base classifiers were considered to enhance the heterogeneity of the proposed ensemble model.

3. Material and Methods

3.1. Datasets

We use power grid datasets developed by the Oak Ridge National Laboratories to verify the effectiveness of benchmark classification models. There are 128 features in total in the power grid datasets. Such features were derived from four Phasor Measurement Units (PMUs), which estimate electrical pulses in a junction utilizing a standard time reference to guarantee proper time synchronization. The last feature denotes the marker that distinguishes between normal and attack occurrences. Around 5000 examples are included in each collection, with 294 examples of no occurrences, 1221 samples of natural occurrences, and 3711 samples of malicious occurrences, respectively, indicating that the provided datasets are imbalanced in nature (see Table 1). In this study, a total of fifteen binary datasets are considered, where each dataset is labeled with a letter from a to o. In addition, we estimate the imbalance ratio (IR) for each dataset, which is defined as a proportion of examples in the minority classes to majority classes. Consequently, highly skewed datasets have a low IR, and vice versa. We consider an IR > 0.4 criteria to indicate datasets with less severe imbalance. The IR value for each dataset is depicted in Figure 1.

Table 1. The characteristics of the power grid datasets w.r.t. the number of samples that belong to attack or natural class.

	Datasets														
Class	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**
# of natural examples	1100	1544	1604	1800	1481	1477	1326	1544	1770	1648	1282	1771	1153	1353	1861
# of attack examples	3866	3525	3811	3402	3680	3490	3910	3771	3570	3921	3969	3453	4118	3762	3415

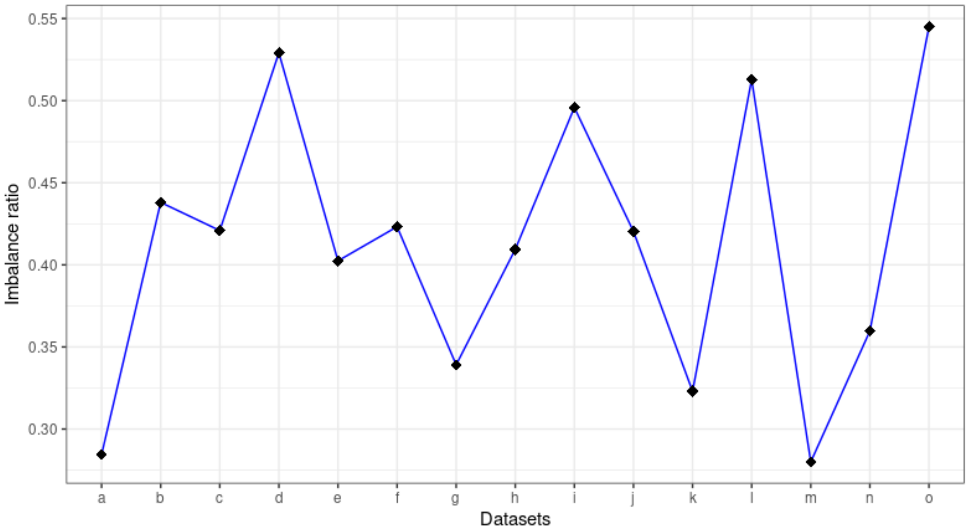


Figure 1. A plot illustrating imbalance ratio for each dataset in the power grid datasets. With the exception of dataset a, g, k, m, and n, all other datasets exhibit a less severe imbalance (e.g., IR > 0.4 in our case).

3.2. Methods

This study includes the implementation of nine class imbalance learning models in *R* [18]. The following section summarizes all classifiers. Each classifier is grouped by the family to which it belongs. Throughout the experiment, all hyperparameters were searched in order to find the optimal ones. Unless otherwise specified, the default learning settings are utilized.

- Cost-sensitive tree learner.
  We apply a pruning procedure and a minimum number of samples for splitting is 2. Additionally, cost ratio is determined by the IR of each dataset.
  - Cost-sensitive decision tree (CSC4.5) [6]. It is a cost associated with misclassifying minority occurrences and is used to alter the training set's class distribution in order to coerce cost-sensitive trees. This mechanism often results in a reduction in the overall misclassification cost, high-cost misclassification, and the size of the tree in a binary prediction task. When training datasets contain a reasonably skewed distribution of classes, smaller trees are a logical byproduct of the tree induction technique.
- Cost-sensitive bagging.
  For the following bagging-based ensemble techniques, we used decision tree (e.g., J48) as a base classifier and set the number of bags to 100. Parallel technique is used to accelerate the training process.
  - RBBagging [7]. Compared to conventional bagging-based approaches for imbalanced data, which employ the same amount of dominant and less dominant samples for each training subset, individual models in RBBagging exhibit a greater degree of diversity, which is one of the essential characteristics of ensemble models. Additionally, the suggested strategy fully utilizes all minority examples through under-sampling, which is accomplished quickly through the use of negative binomial distributions.
  - ROSBagging and RUSBagging [8]. Each subset S_k is constructed in ROSBagging by randomly oversampling minority classes in order to form the k-th classifiers. Similarly, RUSBagging creates each subset by randomly under-sampling majority classes. A majority vote is applied when a new instance occurs after construction and each classifier votes. The class with the most votes ends up making the final decision.
  - SMOTEBagging [8]. It entails the production of synthetic instances as part of the subset formation process. According to SMOTE, two components must be determined: the number of nearest neighbors k and the extent to which the less majority class is over-sampled.
- Cost-sensitive boosting.
  A large number of trees (100) is employed as a base learner, and the decision tree method is applied. The cost ratio is varied according to the IR of each dataset.
  - RUSBoosting [9]. It combines the Adaboost algorithm and random undersampling technique, which reduces examples from the dominant class at random intervals once that sufficient balance is attained.
  - SMOTEBoosting [10]. It incorporates SMOTE into the Adaboost algorithm. SMOTE creates new minority samples by extrapolating previous samples. SMOTEBoosting exacerbates higher model training time limitation as SMOTE is a more sophisticated and time-consuming data sampling strategy.
  - AdaC2 [11]. Introducing the cost items into the AdaBoost by incorporating the cost items into the weights update approach, AdaC2 directs the learning towards instances of the marginalized class. It thus increases the effectiveness of the learning framework. The resampling strategy of AdaC2 is based on the principle that false negatives gain more weight than false positives, whereas true positives lose more weight than true negatives.
- Cost-sensitive hybrid ensemble.
  Similarly, 100 decision trees were used to construct the ensemble, while the cost ratio varies depending on the dataset's IR.

– EasyEnsemble [12]. It employs a two-stage ensemble technique that incorporates random undersampling during bagging iterations and trains the base classifier using AdaBoost. It trains the base classifier by doing bagging in an unsupervised manner.

4. Result and Discussion

In this study, we used nine cost-sensitive learners to analyze fifteen power grid-based intrusion datasets, resulting in a total of 135 classifier-dataset combinations. In order to avoid bias results due to performance results that could be acquired by chance, we used stratified multiple hold-outs (also known as subsampling). Subsampling is carried out by several runs (ten times in our case), partitioning the data set D into a training and a test set by a predetermined proportion, e.g., 75% training and 25% test set in our case. This stratified process allows us to keep the IR of the original dataset. The results of the test are the mean of ten different elements. Since we dealt with the imbalanced classification problem, we adopted three different metrics to measure the classifiers' performance: balance accuracy, Cohen's kappa, and area under ROC curve (AUC). These metrics are calculated in the following ways:

$$\text{Balanced accuracy} = \frac{TPR + TNR}{2} \tag{1}$$

$$Kappa = \frac{2 \times (TP \times TN - FN \times FP)}{(TP + FP) \times (FP + TN) + (TP + FN) \times (FN + TN)} \tag{2}$$

$$AUC = \int_0^1 \frac{TP}{TP + TN} d \frac{FP}{FP + FN} \tag{3}$$

The average performance of all classifiers over the benchmarked datasets is depicted in Figure 2. CSC4.5 exhibits greater variability in performance compared to other algorithms. This result is similar to that of [19], where the cost-sensitive tree such as CSC4.5 should be designed not only by considering the misclassification cost but also by considering the cost of the tests. Furthermore, our experiment revealed that EasyEnsemble is the top performer across all validation metrics, while AdaC2 is the worst performer in terms of balanced accuracy and kappa metrics. The results are pretty reasonable since EasyEnsemble is built using a two-level ensemble which is more advantageous than other standard ensemble models.

We run hierarchical clustering of classifiers and datasets based on their average performance value (see Figure 3). An Euclidean distance and Ward's clustering criterion were utilized to carry out the clustering task. Our study reveals three different clusters of classifiers and datasets. The clusters of classifiers are generally consistent with the original groups, such as bagging and boosting-based ensembles. However, the performance of AdaC2 does not appear to belong to any clusters when it is measured by balanced accuracy and kappa metrics. On the contrary, AdaC2 is somewhat comparable to other boosting and hybrid-based ensemble models w.r.t. AUC metric. The clusters also revealed significant performance differences across power grid datasets. For instance, the first cluster (e.g., red dendrogram) consists of the highly imbalanced datasets with low IR score, while, on the other hand, the third cluster (e.g., blue dendrogram) is comprised of less severe imbalanced datasets.

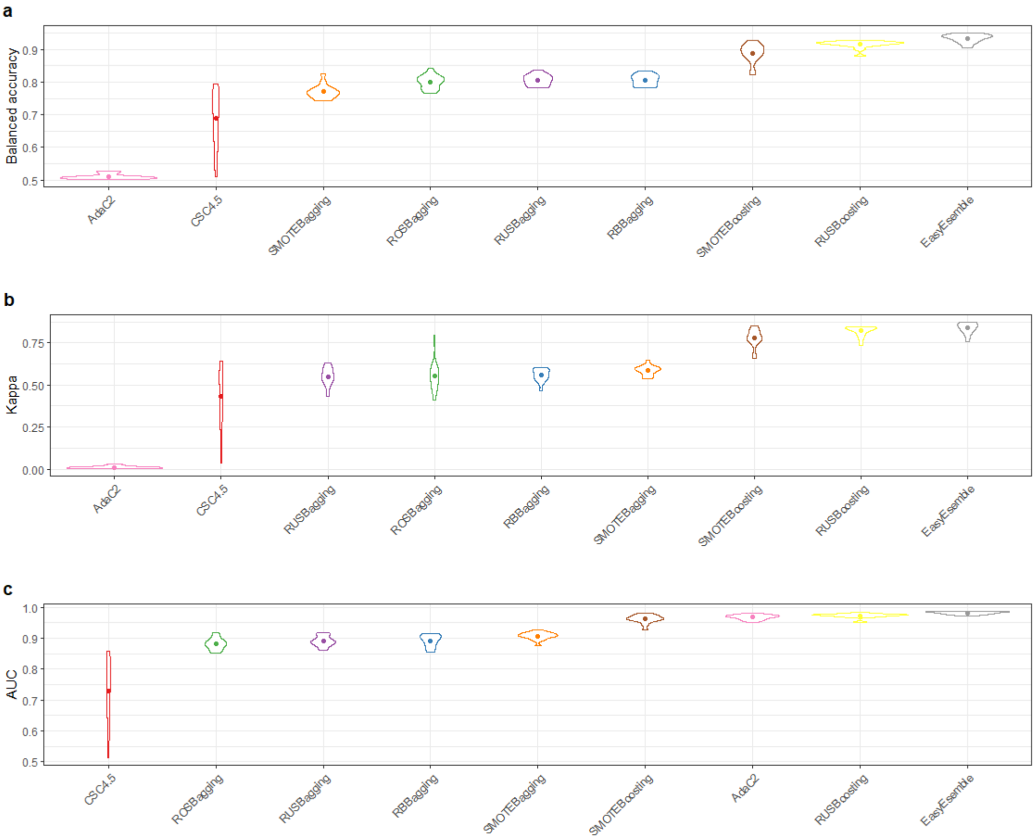


Figure 2. Violin plot showing a full performance distribution of classifiers as measured by balanced accuracy (**a**), Cohen's kappa (**b**), and AUC (**c**) metrics. Classifiers are arranged in descending order based on their mean value (shown as dots in the figure).

Consecutively, we assess the performance differences of classifiers across datasets based on the Nemenyi test. When the corresponding average ranks of two classifiers differ by at least the critical difference (CD), two classifiers are considered significantly different. Figure 4 confirms that EasyEnsemble is on the top rank, followed by RUSBoosting, and SMOTEBoosting. The performance of the top three algorithms is significantly different from any other algorithms in terms of balanced accuracy and kappa metrics. For overall assessment, the performance of EasyEnsemble is rather predictable, since it is a two-level ensemble method that involves bagging and boosting. In addition, our experimental findings confirm that oversampling strategies perform well in boosting-based ensemble, despite having the risk of adding extra noise to the training data sets. In similar fashion, introducing an undersampling strategy to boosting ensemble demonstrates an advantage, aside from its self-evident issue: by excluding examples from the dominant class, the classifier may overlook essential concepts associated with the dominant class. The performance difference of AdaC2 in various metrics implies that AUC is not necessarily the best metric for evaluating classification performance for the IDS data problem, particularly anomaly-based IDS that demands a low false alarm rate [20,21].

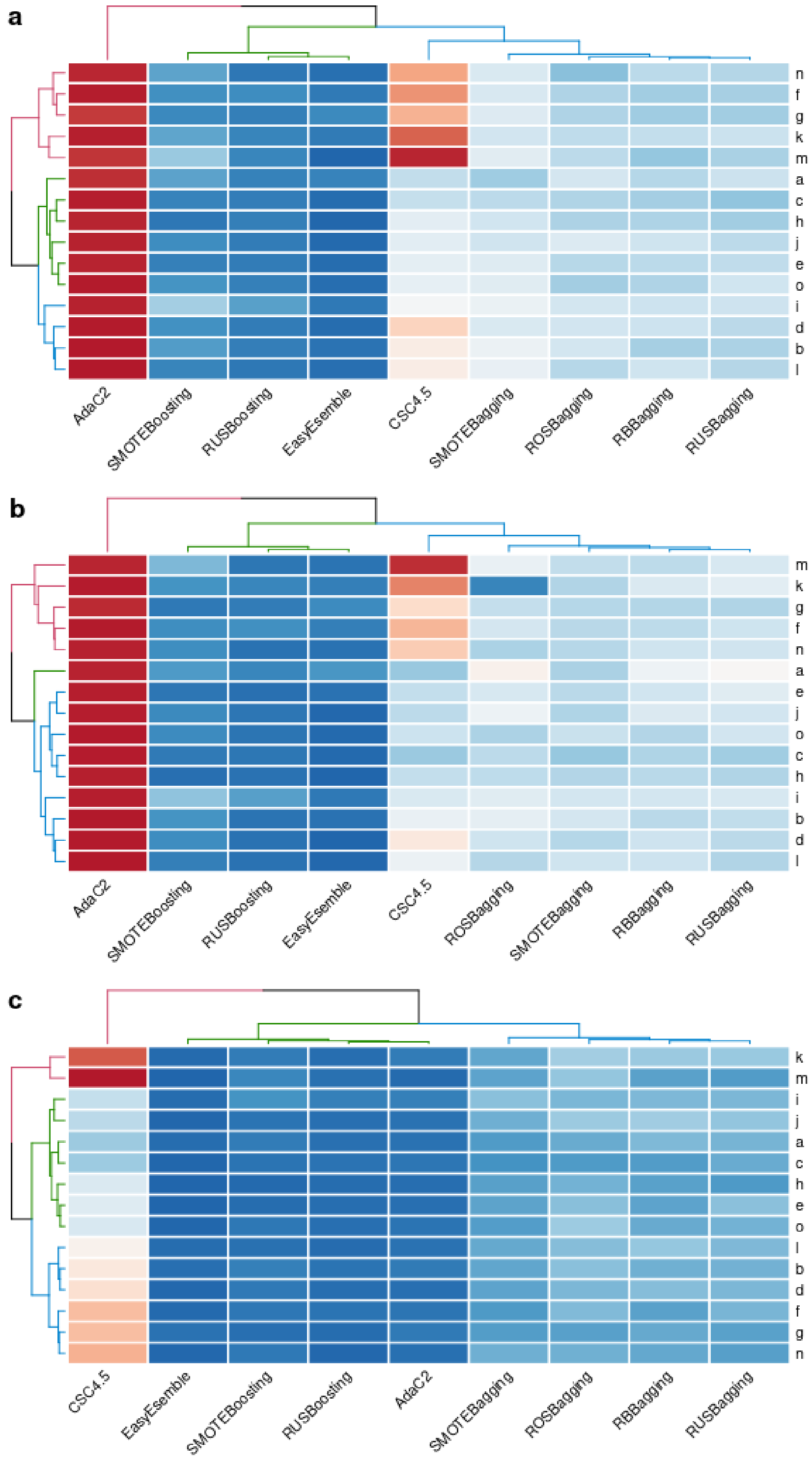
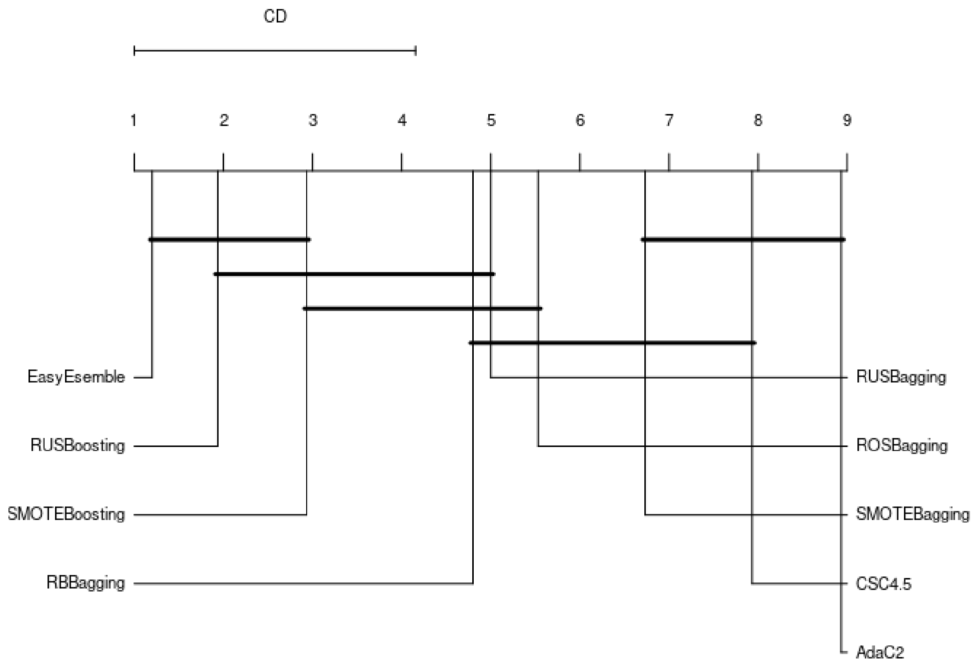
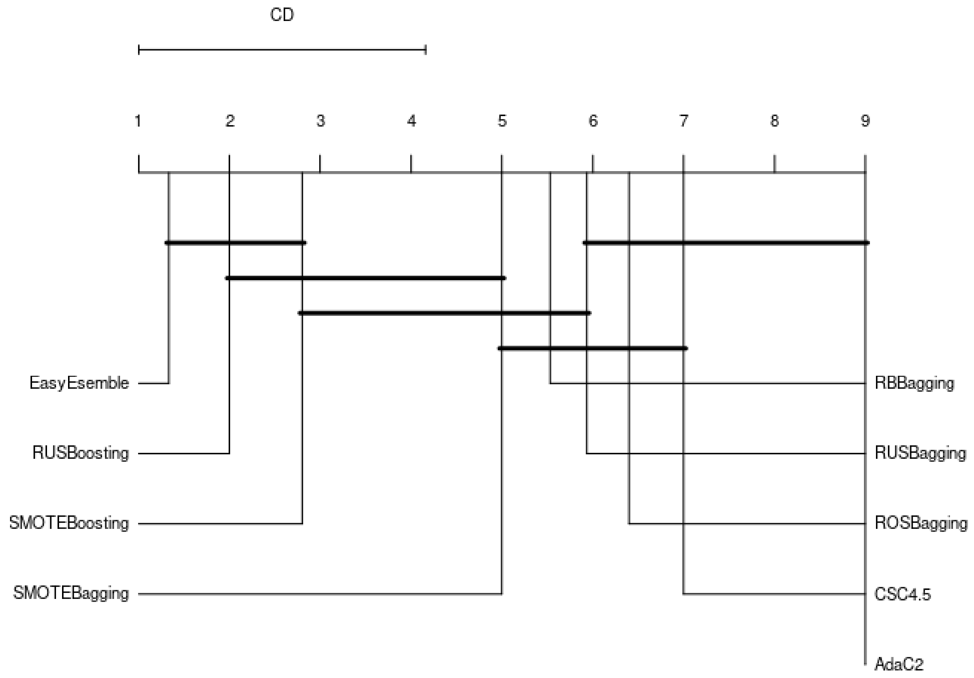


Figure 3. Hierarchical clusters of classifiers and datasets in terms of balanced accuracy (**a**), kappa (**b**), and AUC (**c**) metrics. The clusters are shown by three colors of dendrogram branches.

Finally, we report each algorithm's complexity concerning the time required for training and testing. Note that training and testing time were captured at one time hold-out. As shown in Figure 5 and Table 2, CSC4.5 took a larger amount of time to complete the training but a shorter amount of time during the testing. The result is rather intuitive, as during the training stage of a dataset with n samples and d features, CSC4.5 is required to sort the data and determine the right threshold, which takes $O(n \times log(n) \times d)$. The classifier is quite fast (e.g., $O(nodes)$) during the testing stage as it only identifies the nodes that are normally stored as if-then rules. For overall comparison, despite outperforming other algorithms, EasyEnsemble took approximately 2.7 s to complete the testing task. EasyEnsemble is not the best choice, given that IDS models typically require a real-time detection approach. We suggest that RUSBoosting may be a viable method for implementing IDS in real-world industrial control networks settings.

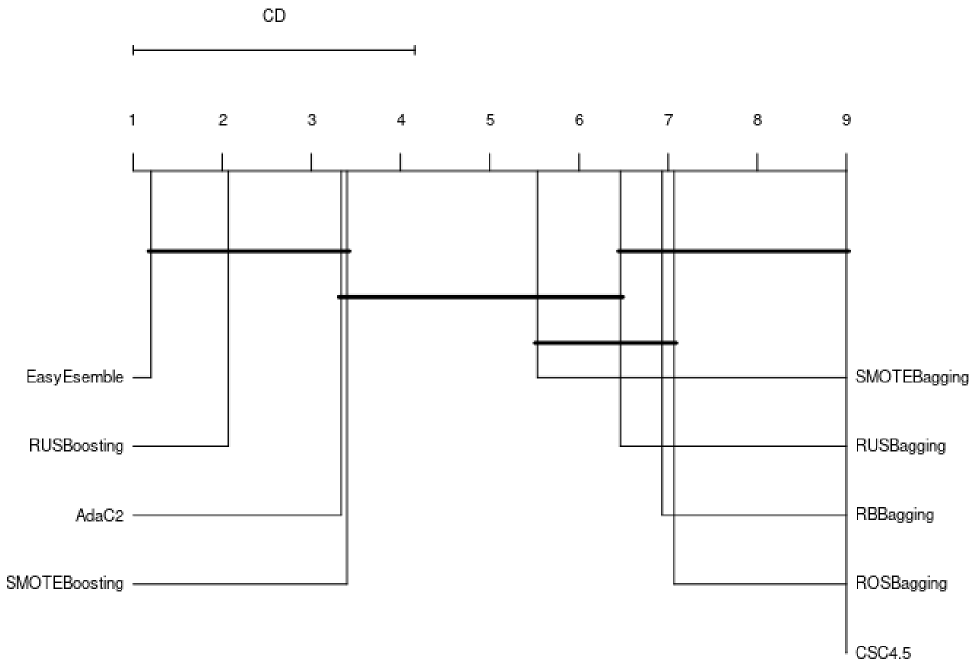


(**a**)



(**b**)

Figure 4. *Cont.*



(**c**)

Figure 4. Comparison of the average performance values such as balanced accuracy (**a**), kappa (**b**), and AUC (**c**), across all datasets attained by all classifiers using the Nemenyi critical difference plot.

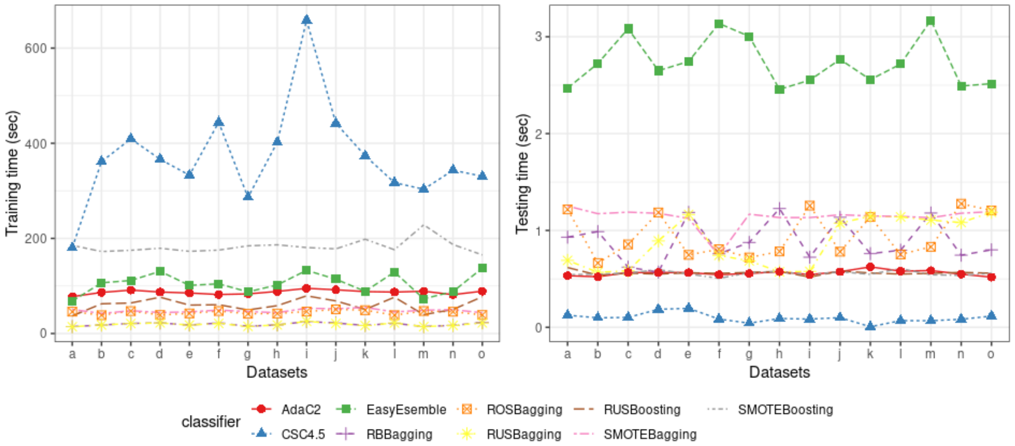


Figure 5. Time required by each classifier in the training and testing stage.

Table 2. Average training and testing time required for each algorithm measured in seconds.

Algorithm	Training Time	Testing Time
CSC4.5	370.09	0.097
RBBagging	19.39	0.887
ROSBagging	44.06	0.948
RUSBagging	19.16	0.881
SMOTEBagging	47.88	1.137
RUSBoosting	60.74	0.562
SMOTEBoosting	183.01	0.557
AdaC2	86.92	0.560
EasyEnsemble	105.10	2.736

5. Conclusions

This study filled the research gap by reporting the pros and cons of cost-sensitive ensemble-based classifiers for classifying and detecting cyberattacks in industrial control networks. Nine cost-sensitive classifiers and fifteen power grid datasets were utilized in the experiment. Despite being faster than the competitors, EasyEnsemble still took the most time throughout the testing stage, leading us to assume that this classifier is unsuitable for this application domain. Additionally, we identify the limitation of our study. This study concentrated on a small number of classifiers and datasets, making it unable to provide a complete validation. We believe that expanding the number of classifiers and datasets would be interesting for future works.

Author Contributions: Conceptualization, M.H.L.L. and B.A.T.; methodology, B.A.T.; validation, M.H.L.L.; investigation, M.H.L.L.; writing—original draft preparation, M.H.L.L. and B.A.T.; writing—review and editing, M.H.L.L. and B.A.T.; visualization, B.A.T.; supervision, B.A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Institute for Basic Science (IBS) under grant No. IBS-R029-C2-001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Zhou, Z.H. *Ensemble Methods: Foundations and Algorithms*; CRC Press: Boca Raton, FL, USA, 2012.
2. Rokach, L. Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Comput. Stat. Data Anal.* **2009**, *53*, 4046–4072. [CrossRef]
3. Kuncheva, L.I. *Combining Pattern Classifiers: Methods and Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
4. Tama, B.A.; Lim, S. Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. *Comput. Sci. Rev.* **2021**, *39*, 100357. [CrossRef]
5. Tama, B.A.; Rhee, K.H. HFSTE: Hybrid feature selections and tree-based classifiers ensemble for intrusion detection system. *IEICE Trans. Inf. Syst.* **2017**, *100*, 1729–1737. [CrossRef]
6. Ting, K.M. An instance-weighting method to induce cost-sensitive trees. *IEEE Trans. Knowl. Data Eng.* **2002**, *14*, 659–665. [CrossRef]
7. Hido, S.; Kashima, H.; Takahashi, Y. Roughly balanced bagging for imbalanced data. *Stat. Anal. Data Min. Asa Data Sci. J.* **2009**, *2*, 412–426. [CrossRef]
8. Wang, S.; Yao, X. Diversity analysis on imbalanced data sets by using ensemble models. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Data Mining, Paris, France, 11–15 April 2009; pp. 324–331.
9. Seiffert, C.; Khoshgoftaar, T.M.; Van Hulse, J.; Napolitano, A. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2009**, *40*, 185–197. [CrossRef]
10. Chawla, N.V.; Lazarevic, A.; Hall, L.O.; Bowyer, K.W. SMOTEBoost: Improving prediction of the minority class in boosting. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, Cavtat-Dubrovnik, Croatia, 22–26 September 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 107–119.
11. Sun, Y.; Kamel, M.S.; Wong, A.K.; Wang, Y. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognit.* **2007**, *40*, 3358–3378. [CrossRef]
12. Liu, X.Y.; Wu, J.; Zhou, Z.H. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B* **2008**, *39*, 539–550.
13. Alimi, O.A.; Ouahada, K.; Abu-Mahfouz, A.M.; Rimer, S.; Alimi, K.O.A. A Review of Research Works on Supervised Learning Algorithms for SCADA Intrusion Detection and Classification. *Sustainability* **2021**, *13*, 9597. [CrossRef]
14. Shahraki, A.; Abbasi, M.; Haugen, Ø. Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103770. [CrossRef]
15. Anton, S.D.D.; Sinha, S.; Schotten, H.D. Anomaly-based intrusion detection in industrial data with SVM and random forests. In Proceedings of the 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 19–21 September 2019; pp. 1–6.
16. Khan, I.A.; Pi, D.; Khan, Z.U.; Hussain, Y.; Nawaz, A. HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems. *IEEE Access* **2019**, *7*, 89507–89521. [CrossRef]

17. Upadhyay, D.; Manero, J.; Zaman, M.; Sampalli, S. Intrusion Detection in SCADA Based Power Grids: Recursive Feature Elimination Model with Majority Vote Ensemble Algorithm. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2559–2574. [CrossRef]
18. Zhu, B.; Gao, Z.; Zhao, J.; vanden Broucke, S.K. IRIC: An R library for binary imbalanced classification. *SoftwareX* **2019**, *10*, 100341. [CrossRef]
19. Lomax, S.; Vadera, S. An empirical comparison of cost-sensitive decision tree induction algorithms. *Expert Syst.* **2011**, *28*, 227–268. [CrossRef]
20. Jeni, L.A.; Cohn, J.F.; De La Torre, F. Facing imbalanced data–recommendations for the use of performance metrics. In Proceedings of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, Geneva, Switzerland, 2–5 September 2013; pp. 245–251.
21. Luque, A.; Carrasco, A.; Martín, A.; de las Heras, A. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognit.* **2019**, *91*, 216–231. [CrossRef]