IoT Network Intrusion Detection with Ensemble Learners

Sulyman Age Abdulkareem*, Chuan Heng Foh*, Haeyoung Lee*, François Carrez*, Klaus Moessner*†

*5GIC & 6GIC, Institute for Communication Systems (ICS), University of Surrey, Guildford, Surrey, U.K.
†Faculty of Electronics and Information Technology, Technical University Chemnitz, Germany.
Email:{s.abdulkareem, c.foh, Haeyoung.Lee f.carrez, k.moessner}@surrey.ac.uk, klaus.moessner@etit.tu-chemnitz.de

***Abstract*—Protecting information systems against intruders' attacks requires utilising intrusion detection systems. Over the past several years, many open-source intrusion datasets have been made available so that academics and researchers can analyse and assess various detection classifiers' effectiveness. These datasets are made available with a full complement of illustrative network features. In this research, we investigate the issue of Network Intrusion Detection (NID) by utilising an Internet of Things (IoT) dataset called Bot-IoT to evaluate the detection efficiency and effectiveness of five different Ensemble Learning Classifiers (ELCs). Our experiment's results showed that despite all ELCs recording high classification metric scores, CatBoost emerged as the ELC that performed the best in our experiment in terms of Accuracy, Precision, F1-Score, Training and Test Time.**

***Index Terms*—Network Intrusion Detection, Machine Learning, Ensemble Learning Classifiers, CatBoost, IoT.**

I. Introduction

Our everyday lives are becoming increasingly intertwined with vast amounts of data thanks to the fast expansion of information technology. Cisco has projected that IP traffic is expected to expand from 120 exabytes per month in 2017 to 400 exabytes per month in 2022 [1]. Increased network traffic has led to a growth in the amount of cyberattack-related risks, which have become more diverse. The word "cyberattack" is often used to describe an uninvited attempt to threaten, disable, damage, steal, or otherwise compromise another party's information assets. Many businesses now rely on network intrusion detection systems (NIDS) to keep their networks safe. Security measures such as firewalls, virus protection, data encryption, and user authentication are essential but not sufficient to protect computers and networks from today's threats. In the face of these issues, intrusion detection systems (IDS), a Machine Learning (ML) based method, and the aforementioned security measures can work together [2].

An IDS monitors and analyses network traffic in real-time to detect latent data anomalies. IDSs may be divided into two categories based on their detection philosophy [3]. The first one is Signature-based intrusion detection which uses predefined attack signatures to characterise intrusion attempts on the network. As such, this approach cannot detect new attacks [4]. On the other hand, anomaly-based detection may uncover previously undisclosed attacks by analysing network data for anomalies using machine learning algorithms. An anomaly is an incident or behaviour that is out of the ordinary. Many studies [5]–[8] have focused on enhancing the accuracy and efficiency of IDSs. Anomaly-based IDS has been widely implemented and is now the primary focus of IDS research due to its promising efficacy.

In recent years, machine learning algorithms such as Decision Tree, Random Forest, Support Vector Machine (SVM), and Neural Networks have been applied to intrusion detection. However, each algorithm has its advantages and downsides. Classifiers that work well to detect a particular attack may not work well for another. According to several previous research publications [9]–[11], there are still certain drawbacks, no matter the pre-processing or feature selection methods used alongside the classifiers. The ML architecture for IDS continuously grows into increasingly complex classifiers to increase its efficacy. Ensemble Learning Classifiers (ELCs) for classification are an example of improving the coherence and competence of detecting intrusions. The approach has become more popular than the use of single classifiers. To compensate for the deficiencies of the weak classifiers, it combines them to build a powerful learner. The ELCs are better options over single classifiers since they produce better performance.

This paper evaluates the performance of five Ensemble Learning Classifiers on the Bot-IoT dataset [12], an IoT dataset to test for multiclass classification performance. CatBoost, Random Forest, LightGBM, and XGBoost are the ELCs employed in this investigation. The two primary focuses of this study are detection effectiveness and speed. Smart homes, smart cities, and smart transportation systems are among the many applications for the Internet of Things (IoT) we looked at in our research. However, IoT security protection is still lacking compared to traditional network applications. We test the detection abilities of the five ELCs on the IoT dataset's network categories [12]. Our experiment findings indicate that CatBoost is able to give the best classification result compared to other ELCs. CatBoost also trained and tested the dataset in the shortest time, making it a superior ELC over the other ELCs in our study.

The remainder of this paper is organised as follows. Section II overviews some of the most recent NID ensemble ML classifier experiments. Section III introduces our experiment IoT dataset, Ensemble Learning Classifiers, as well as our research strategy and experiment design. Comparing the performance outcomes among ELCs is the focus of Section IV. The latter part of Section V consists of some critical observations and concludes this study.

II. Related Works

The idea behind the 'Ensemble Learning' approach is to combine data mining and machine learning into one single process [13]. For classification, this study used ensemble learners, a collection of individual learning techniques known as weak learners. Classifiers with little learning potential are brought together and taught as a group to get better classification results [14]. Most academics and researchers working on ML-related research are now employing ELCs in their work as they have shown more remarkable results than single learning classifiers. The capacity of the base learning techniques to deliver increased safety through enhanced intrusion detection may be improved by utilising ensemble learning methods. Verma *et al.* [15] conducted research comparing the effects of applying single learning and ensemble learning classifiers using open datasets. When compared to the results of the single learning classifiers, the ensemble learners produced much better results. Nevertheless, the purpose of this study is to conduct a comparative analysis by making use of a diverse range of ELCs. In the following, we review some state-of-the-art studies that have applied ELCs on network datasets in their work.

Bansal and Kaur [16] analysed XGBoost-based tuning for classification in NID. A robust and efficient intrusion detection classifier, XGBoost, was employed. XGBoost, AdaBoost, NB, MLP, and kNN classifiers for binary and multiclass classification were also tested using the CICIDS2017 dataset Distributed Denial of Service (DDoS attack types only). Using the XGBoost classifier, 91.36% and 99.54% of binary classes were correctly classified. However, the average class error rate for both classifications was high.

Six supervised ML classifiers for multiclass classification were examined by Obeidat *et al.* [17] to evaluate their ability to detect attacks on the KDD99 dataset. Only 60,000 randomly generated KDD99 test sets were utilised in the investigation. Random Forest was more effective in the classification investigation than the other options, with 93.78% accuracy. J48, a variant of the Decision Tree classifier, had a precision of 93.11%, whereas Random Tree had a precision of 90.58%.

Larriva-Novo *et al.* [18] investigated the performance of a set of single learners on the UNSW-NB15 dataset, from which seven best-performing learners are used to form the base learners. The learners are combined using XGBoost as the meta-learner for the final classification. In the evaluation, Synthetic Minority Oversampling Technique (SMOTE) is also used to balance the dataset records, and the correlation Kendall coefficient feature selection technique is applied to remove redundant features. The study shows improved performance after dataset records are balanced using SMOTE.

Using the NSL-KDD dataset, Rajadurai and Gandhi [19] aimed to demonstrate that ensemble classifiers can successfully identify network attacks. Gradient Boosting and RF classifiers form the base learner of the proposed ensemble classifier. As a result, the ensemble classifier correctly classified 91.16% of the instances. In other network categories, however, recall and detection rates were poor, decreasing the classifier's ability to detect anomalies.

A stacked ensemble learner with feature selection technique is proposed by Shi *et al.* [20]. A fusion of two classifiers, namely Extreme Tree Classifier and QDA, is used to obtain the learning result, and KDD99 as well as NSL-KDD datasets are used for the testing. Their experiment results show that the proposed learner maintained a stable performance on both datasets and higher accuracy than other classifiers. Additionally, the development time of the learner is improved due to the application of the feature selection. However, non-IoT datasets were used in evaluating the ELC.

Non-IoT dataset classification has been the primary focus of much of the work, with many algorithms reporting significant false-positive rates and computation times. In some research, neither the classification task type nor the experiment duration is specified. In addition, we discovered that most research evaluating the effectiveness of various ML classifiers on the Bot- IoT dataset employed the binary or 5-class category [21]–[23]. In this work, we apply multiple ELCs to classify the Bot-IoT dataset instances to assess their classification effectiveness alongside their training and test time.

III. IoT Network Intrusion Detection based on Ensemble Learner

In the following, we shall introduce the experimental dataset and the ELCs[1]. Data description, dataset preparation, and experiment setup shall be covered in this section.

A. The Dataset and Pre-processing

We shall use the Bot-IoT dataset [12] created by Koroniotis *et al.* for our experimentation. The dataset has four *csv* files that hold the training and test sets. The instances of the dataset have the labels ”attack” (two network classes), ”category” (five network classes), and ”subcategory” (11 network classes). In addition, 43 network features are contained in the dataset. Table I illustrates the distribution of the dataset instances utilised in this experiment.

The Bot-IoT dataset was compiled in the Cyber Range Lab at UNSW Canberra using real and simulated IoT network traffic and various attacks. To accomplish this, a realistic testbed environment with typical and variant botnet anomalies was designed to collect extensive network data (Denial of Service, Distributed Denial of Service, Information Gathering and Information Theft). Data Exfiltration (DE), DoS-HTTP (DH), DDoS-HTTP (DDH), Keylogging (KL), OS Fingerprint (OSF), and Service Scan (SS) were the subcategory anomaly types of the variants. The bulk of network data consists of DoS-UDP (DU), DoS-TCP (DT), DDoS-UDP (DDU), and DDoS-TCP (DDT), whilst the remainder consists of all other network data.

[1] The source code is available at https://github.com/cfoh/IoT-Network-Intrusion-Detection-with-Ensemble-Learners

TABLE I
SUMMARY OF THE DATASET INSTANCES

Category	Subcategory	Instances
Denial of Service (DoS)	DU	1032961
	DT	615800
	DH	1485
Distributed DoS	DDU	576876
	DDT	348751
	DDH	989
Information Gathering	SS	64280
	OSF	17780
Information Theft	KL	73
	DE	6
Normal	Normal	477
Class Distribution Attack		Normal: 477 (0.02%) Anomaly: 2,659,001 (99.98%)
Total Number of All Instances		2,659,478

We noticed that not all features were essential for network classification during preprocessing. *flgs*, *daddr*, *pkSeqID*, *proto*, *saddr* and *state* were eliminated in particular. We observed that *flgs*, *proto* and *state* store the same information as *flgs number*, *proto number* and *state number*, respectively. *daddr*, *pkSeqID* and *saddr* were eliminated since they are device-specific.

The subcategory name is converted to integer values ranging from 0 to 10, encompassing all 11 network categories, as the scope of our study includes all network dataset instances. Additionally, the dataset was split into 80 percent and 20 percent, as in the work of Churcher *et al.* [24], eighty percent of the data is utilised for training, while the remaining twenty percent is used to test the classifiers. We also do min-max scaling normalisation to address skewness in the features by scaling to the range of 0 to 1.

B. Ensemble Learning Classifiers (ELCs)

The basic principle of the ensemble learning classifiers is to aggregate weak learners to create a strong learner [25]. The ELC uses a two-tier classification approach, with base learners classifying cases at the first level. The meta-learner then identifies and learns the outputs of the base learners. Before giving the final classification, the second-level classifier resolves the losses of the previous level [26]. Bagging, Boosting, and Stacking are three classification techniques based on ensemble learning. In this work, we focus on variants of bagging and boosting ELCs. A brief overview of some popular ELCs is given as follows.

1) AdaBoost: AdaBoost [27] is an iterative classifier that combines many weak classifiers into a single robust classifier. This classifier's central concept is to train many weak classifiers on the same training data. This classifier modifies the sample weight based on the outcome of each training and the accuracy of the last overall classification and then trains the next weak classifier with the new data. AdaBoost computes the accuracy of the weak classifiers and combines them into a robust classifier for the final decision. When a given set of conditions are satisfied, the iterative process ends.

2) LightGBM: LightGBM [28] is a Gradient-Boosted Decision Trees (GBDT) classifier incorporating gradient-based one-Side sampling (GOSS) and mutually exclusive feature bundling (EFB). The earlier GBDT classifier has a more extended training period, with establishing the appropriate split point accounting for most of the time. LightGBM employs the histogram approach for feature selection and segmentation point determination to address this issue. This approach bins the original continuous feature values and builds the classifier using these bins. The histogram drastically decreases the time required to choose split points and enhances a classifier's training and prediction efficiency.

3) Random Forest: Random Forest (RF) [29] is an ensemble of untrimmed classification or regression trees. It is currently the most accurate data mining method, especially for massive datasets with several attributes. The random forest produces several classification trees. Using a tree classification classifier, a separate bootstrap sample from the original data is used to generate each tree. After the forest has been created, a new item requiring classification is placed on each tree. Each tree casts a vote indicating its decision on the instance's class. The forest selects the class with the highest number of votes for an instance for its final classification decision.

4) CatBoost: CatBoost [30] is an open-source machine learning library created in 2017 by the Russian search engine Yandex. As with the well-known XGBoost and LightGBM, it belongs to the Boosting family. It has a rapid learning rate and performs well with numeric, category, and textual data. CatBoost overcomes the problems of gradient bias and prediction shift in the Boosting family classifier, improving prediction accuracy and partially resolving the overfitting issue. In addition, CatBoost can efficiently and adequately analyse discrete data. Unlike previous classifiers, it has GPU support, and visualisation features.

5) XGBoost: XGBoost [31] was created in 2014 by Tianqi Chen, and it is a GBDT version that has been tweaked to boost speed and prediction performance. It is a scalable approach compatible with R, Python, Hadoop, Scala, and Julia. XGBoost includes a number of parameters that decrease overfitting and boost overall performance. Thus, it delivers precision, practicability, and efficacy. It can run automatically in parallel on Windows and Linux and is up to ten times quicker than conventional GBDT.

All experimental simulations were conducted on a 64-bit Windows 11 computer using Python. Other PC features include 8GB RAM and an Intel Core i7-8550U processor running at 1.80 GHz. In addition, the ML classifiers were created with the Scikit-Learn toolkit. This was done in the Anaconda Navigator GUI environment using the Jupyter Notebook IDE.

IV. PERFORMANCE EVALUATION

This section discusses the detection performance of the ELCs implemented in our experiment. Five ELCs were investigated in this paper. The investigation was accomplished using Accuracy, Precision, Recall, F1-Score, Training, and Test Time.

TABLE II
COMPREHENSIVE COMPARISON BETWEEN ALL THE ELCS (METRICS ARE IN %)

Classifier	Accuracy	Precision	Recall	F1	Train (sec)	Test (sec)
AdaBoost	99.91	81.77	81.56	81.66	735.21	14.10
LightGBM	96	44	43	43	575.40	66.29
RF	98.94	52.22	53.11	52.63	286.11	9.28
CatBoost	99.99	99.89	99.73	99.81	229.42	1.84
XGBoost	99.99	99.70	99.84	99.77	4979.86	24.74

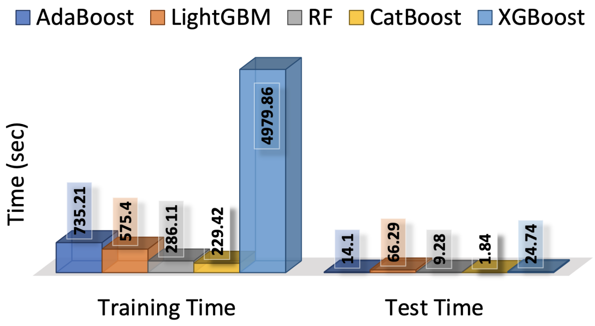

Fig. 1. Time comparison between the 5 ELCs (metrics are in *sec*).

We evaluated the performance of five ELC machine learning classifiers on an IoT dataset to determine their efficacy and detection speed. Table II summarises the performance of all the five ELCs. The results revealed that all the evaluated classifiers recorded over 95% overall classification accuracy, indicating that they could classify most of the network instances correctly.

The CatBoost algorithm has the best training time, whereas Random Forest needs more than four minutes for training. The training period for the XGBoost classifier is around one hour and forty-eight minutes, whereas the prediction test takes only 25 seconds. In terms of training time, we conclude that the XGBoost classifier is the slowest, and the CatBoost classifier is the quickest. The inference time or prediction test time is crucial, as intrusion detection systems often operate in real-time. A classifier with a lengthy prediction time will impede the overall network's performance. However, the CatBoost and XGBoost classifiers reach the highest level of accuracy, far surpassing the LightGBM. The ELC with the worst performance in terms of classification metrics is LightGBM, which had the lowest score for every measure except training time. Regarding training and evaluation time, CatBoost outperformed the other classifiers in our experiment. Thus, we choose CatBoost as the best classifier for the IoT dataset due to its fast training and testing times and superior predicted accuracy.

Considering the overall results, CatBoost ELC is more effective and efficient than state-of-the-art ELCs in detecting IoT network intrusions. Some of the advantages of the classifier are its robustness, reducing the need for extensive hyper-parameter tuning, lowering the chances of overfitting, and providing state-of-the-art classification results. In addition, it has a reduced pre-processing time as it can handle categorical features automatically. These advantages explain why the classifier can deliver the best classification result over other ELCs in our experiment. However, we believe that removing irrelevant features from the output variable should reduce the training and test time while maintaining the same level of detection performance.

V. CONCLUSION

This study examined the performance of five ensemble learning classifiers for anomaly detection on an Internet of Things dataset. According to our findings, CatBoost outperformed four other ELCs for the eleven multiclass network categories. CatBoost had the highest overall classification metric scores and the shortest training and testing time among all five ELCs. To further maximise the classification performance, we suggest advancing this work by experimenting with different feature selection techniques on the dataset to evaluate further the trade-off between training and testing computing resource requirements. The five classifiers may also be assessed on other IoT datasets containing more diverse attacks to further evaluate their efficacy and efficiency.

ACKNOWLEDGMENT

This work was also partly sponsored by the Horizon 2020 research and innovation programme of the European Union under the projects L3Pilot (grant agreement number. 723051), 5G-HEART (grant agreement no. 857034), and DEDICAT 6G (grant agreement no. 101016499). The contents of this publication are the sole responsibility of the authors and do not in any way reflect the views of the EU. We would also like to recognise the 5GIC/6GIC members' contribution to this study.

REFERENCES

[1] C. Whitepaper, "Cisco Visual Networking Index: Forecast and Trends, 2017–2022," Cisco, 2018.

[2] S. Pontarelli, G. Bianchi, and S. Teofili, "Traffic-aware design of a high-speed fpga network intrusion detection system," IEEE Transactions on Computers, vol. 62, no. 11, pp. 2322–2334.

[3] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques,systems and challenges, "computers & security, vol. 28, no. 1-2, pp.18–28, 2009.

[4] Y. Tang and S. Chen, "An automated signature-based approach against polymorphic internet worms," IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 7, pp. 879–892, 2007.

[5] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. Hashem, "Attack and anomaly detection in iot sensors in iot sites using machine learning approaches," Internet of Things, vol. 7, p. 100059, 2019.

[6] P. B. Dash, J. Nayak, B. Naik, E. Oram, and S. H. Islam, "Model based iot security framework using multiclass adaptive boosting with smote," Security and Privacy, vol. 3, no. 5, p. e112, 2020.

[7] N. K. Sahu and I. Mukherjee, “Machine learning based anomaly detection for iot network:(anomaly detection in iot network),” in 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184). IEEE, 2020, pp. 787–794.

[8] D. K. K. Reddy and H. Behera, “Catboosting approach for anomaly detection in iot-based smart home environment,” in Computational Intelligence in Data Mining. Springer, 2022, pp. 753–764.

[9] O. Almomani, M. A. Almaiah, A. Alsaaidah, S. Smadi, A. H. Mohammad, and A. Althunibat, “Machine learning classifiers for network intrusion detection system: comparative study,” in 2021 International Conference on Information Technology (ICIT). IEEE, 2021, pp. 440–445.

[10] A. H. Azizan, S. A. Mostafa, A. Mustapha, C. F. M. Foozy, M. H. A. Wahab, M. A. Mohammed, and B. A. Khalaf, “A machine learning approach for improving the performance of network intrusion detection systems,” Annals of Emerging Technologies in Computing (AETiC), vol. 5, no. 5, pp. 201–208, 2021.

[11] S. Latif, F. F. Dola, M. Afsar, I. J. Esha, and D. Nandi, “Investigation of machine learning algorithms for network intrusion detection.” International Journal of Information Engineering & Electronic Business, vol. 14, no. 2, 2022.

[12] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, ”Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset.” *Future Gener. Comput. Syst.* 2019.

[13] M. A. Jabbar and R. Aluvalu, ”RFAODE: A novel ensemble intrusion detection system”, Procedia computer science, vol. 115, pp. 226-234, 2017.

[14] O. Sagi and L. Rokach, ”Ensemble learning: A survey”, Wiley Interdisciplinary Reviews: *Data Mining and Knowledge Discovery*, vol. 8, no. 4, pp. 1-18, August 2017.

[15] P. Verma, S. Anwar, S. Khan, and S. B. Mane, “Network intrusion detection using clustering and gradient boosting,” in 2018 9th International conference on computing, communication and networking technologies (ICCCNT). IEEE, 2018, pp. 1–7.

[16] A. Bansal and S. Kaur, “Extreme gradient boosting based tuning for classification in intrusion detection systems,” in International Conference on Advances in Computing and Data Sciences. Springer, 2018.

[17] I. Obeidat, N. Hamadneh, M. Alkasassbeh, M. Almseidin, and M. AlZubi, “Intensive pre-processing of kdd cup 99 for network intrusion classification using machine learning techniques,” 2019.

[18] X. Larriva-Novo, C. Sanchez-Zas, V. A. Villagra, M. Vega-Barbas, and D. Rivera, “An approach for the application of a dynamic multi-class classifier for network intrusion detection systems,” Electronics, vol. 9, no. 11, p. 1759, 2020.

[19] H. Rajadurai and U. D. Gandhi, “A stacked ensemble learning model for intrusion detection in wireless network,” NEURAL COMPUTING & APPLICATIONS, 2020.

[20] X. Shi, Y. Cai, and Y. Yang, “Extreme trees network intrusion detection framework based on ensemble learning,” in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*. IEEE, 2020, pp. 91–95.

[21] D. D. Kulkarni, S. Rathore, and R. K. Jaiswal, “Intrusion detection system for iot networks using neural networks with extended kalman filter,” in *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2021, pp. 1–7.

[22] H. Tyagi and R. Kumar, “Attack and anomaly detection in iot networks using supervised machine learning approaches.” *Rev. d’Intelligence Artif.*, vol. 35, no. 1, pp. 11–21, 2021.

[23] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, “Federated deep learning for zero-day botnet attack detection in iot edge devices,” *IEEE Internet of Things Journal*, 2021.

[24] A. Churcher, R. Ullah, J. Ahmad, F. Masood, M. Gogate, F. Alqahtani, B. Nour, W. J. Buchanan et al., “An experimental analysis of attack classification using machine learning in iot networks,” Sensors, vol. 21, no. 2, p. 446, 2021.

[25] S. Bagui, and K. Li, ”Resampling imbalanced data for network intrusion detection datasets.” *J. Big Data* 2021.

[26] R. Qaddoura, A. Al-Zoubi, I. Almomani, and H. Faris, ”A Multi-Stage Classification Approach for IoT Intrusion Detection Based on Clustering with Oversampling.” *Appl. Sci.* 2021.

[27] Q. Wang and X. Wei, “The detection of network intrusion based on improved adaboost algorithm,” in Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, 2020, pp. 84–88.

[28] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu LightGBM: a highly efficient gradient boosting decision tree Advances in Neural Information Processing Systems (2017), pp. 3146-3154.

[29] L. Breiman, ”Random Forests”, Machine Learning 45(1):5-32, 2001.

[30] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, M. Asadpour Boosting methods for multi-class imbalanced data classification: an experimental review J Big Data, 7 (2020), p. 70.

[31] 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.