2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService) | 978-1-6654-3483-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigDataService52369.2021.00023

Mitigating Class Imbalance for IoT Network Intrusion Detection: A Survey

Joffrey L. Leevy, Taghi M. Khoshgoftaar, Jared M. Peterson

Email: jleevy2017@fau.edu, khoshgof@fau.edu, jpeterson2019@fau.edu

***Abstract*—As the number of *Internet of Things* (IoT) devices continues to rapidly increase, the need to effectively manage the related security risks has become more obvious. For this reason, datasets such as Bot-IoT were created to train machine learning models on network-based intrusion detection. Bot-IoT is modern, publicly available, and covers a wide range of botnet attack traffic in IoT networks. Out of the roughly 73,000,000 instances contained in this dataset, only about 0.013% represents normal traffic, which indicates that the issue of class imbalance should not be ignored for Bot-IoT. Our contribution includes several important findings based on works that address the imbalance in this dataset. In general, we noted the excellent performance of a diverse range of trained models. This suggests Bot-IoT is a reliable dataset that is relatively easy to classify. We also observed that information on data cleaning was left out in a few papers, thus making it challenging for outside researchers to reproduce experiments from these works. Finally, we noted the popularity of both data-level and algorithm-level approaches for mitigating class imbalance in Bot-IoT.**

***Index Terms*—Bot-IoT, intrusion detection, class imbalance, big data, internet of things, machine learning**

I. Introduction

The *Internet of Things* (IoT) refers to a network of devices, not normally considered computers, that have Internet connectivity and computing capability [1]. As the use of these smart devices rapidly increases in diverse networks (home, industrial, military, etc.), there are growing concerns over associated security risks. Technology company Microsoft revealed there was a 35% increase in IoT attacks between the second half of 2019 and the first half of 2020 [2]. Moreover, business magazine Forbes reported that there will be about 35 billion smart devices online in 2021 and about 75 billion in 2025 [3].

Datasets such as Bot-IoT [4] were designed to train machine learning models on network-based intrusion detection for IoT devices. Bot-IoT contains big data, which exhibits specific properties, such as volume, variability, variety, value, velocity, and complexity. These properties can increase the classification burden for trained models.

Created in 2018 by the University of New South Wales, Bot-IoT provides a realistic representation of botnet and normal traffic. The dataset is publicly available and covers a broad range of attacks, including *denial-of-service* (DoS), *distributed denial-of-service* (DDoS), reconnaissance, and theft. It contains 29 features and 73,370,443 instances, of which 9,543 instances (0.013%) are normal traffic. Table I shows categories and subcategories of network traffic for Bot-IoT.

TABLE I
Bot-IoT: Network Traffic Instances

Category	Subcategory	No. of Instances
DoS	TCP	12,315,997
	UDP	20,659,491
	HTTP	29,706
DDoS	TCP	19,547,603
	UDP	18,965,106
	HTTP	19,771
Reconnaissance	OS Fingerprinting	358,275
	Service Scanning	1,463,364
Theft	Keylogging	1,469
	Data Theft	118
Normal	Normal	9,543

From a binary classification perspective, there is a high class imbalance between attack traffic (majority class) and normal traffic (minority class) for Bot-IoT. Class imbalance is caused by a disproportionate number of majority class instances, and could potentially skew the results of big data analytics. High class imbalance is said to occur in a dataset when the majority-to-minority ratio is between 100:1 and 10,000:1 [5].

Strategies for mitigating class imbalance can be exercised at the data level and/or algorithm level [5]. Common data-level approaches include *random oversampling* (ROS), *random undersampling* (RUS), and *synthetic minority oversampling technique* (SMOTE). Certain feature selection algorithms also remedy class imbalance at the data level. Approaches at the algorithm level are mainly concerned with cost-sensitive techniques.

To the best of our knowledge, this is the first survey that investigates Bot-IoT papers related to the use of techniques for addressing class imbalance. Our exhaustive search for relevant, peer-reviewed works ended on April 30, 2021. The contribution of this survey embraces three key findings. For the most part, we noticed that performance scores were exceptional. This may reflect the high reliability of Bot-IoT and is also an indication that the dataset is relatively easy to classify. Secondly, we discovered that information on data cleaning was omitted in a few papers. Data cleaning pertains to the formatting, modification, and deletion of data to improve dataset usability [6]. Finally, we noted the popularity of both data-level and algorithm-level approaches for mitigating class imbalance in Bot-IoT.

978-1-6654-3483-6/21/$31.00 ©2021 IEEE
DOI 10.1109/BigDataService52369.2021.00023

The remainder of this paper is organized as follows: Section II discusses the surveyed works; Section III analyzes findings and highlights gaps within the body of surveyed works; and Section IV concludes with the main points of this paper and provides guidance for future work.

II. Works Addressing Class Imbalance in Bot-IoT

In this section, we investigate works that aim to boost classification performance by tackling the class imbalance in Bot-IoT. Table II provides an alphabetical listing by author of the papers discussed, along with the best respective performance score(s). All scores obtained from the metrics shown in the table (accuracy, precision, recall, F-measure, *area under the receiver operating characteristic curve* (AUC)) are the best scores in each study for binary classification. We point out that comparisons between scores for separate works of research or separate experiments in the same paper may not be valid. This is because datasets may differ in the number of instances and features, and possibly the choice of computational framework. Furthermore, variations of an original experiment may be performed on the same dataset. However, providing these scores may be beneficial for future comparative research. Table III provides an alphabetical listing by author of the papers discussed, along with the respective technique for addressing class imbalance. Lastly, Table IV shows the same ordered listing by author, as well as the proposed respective model.

Bagui & Li, 2021 [7] (Resampling Imbalanced Data for Network Intrusion Detection Datasets)

To mitigate class imbalance in datasets, the authors focused on five resampling strategies: RUS, ROS, *random undersampling and random oversampling* (RURO), *random undersampling with synthetic minority oversampling technique* (RU-SMOTE), and *random undersampling with synthetic minority oversampling technique* (RU-ADASYN) [16]. A feedforward neural network [17] classifier was trained on six datasets: KDD99 [1], UNSW-NB15 [18], (UNSW-NB17-Ecobee_ Thermostat, UNSW-NB17-Danmini_Doorbell, UNSW-NB17-Philips_ B120N10_Baby_Monitor) [2], and Bot-IoT. Models were implemented on a local machine with Scikit-learn [3] and within a big data framework with the Apache Spark Machine Learning Library (MLlib) [4]. Datasets were split in a training to test ratio of 70:30. With regard to classification, the authors discovered that feedforward networks performed better with Scikit-learn than with Apache Spark. As expected, however, model training was much faster with Apache Spark. For Bot-IoT, the best results were obtained with ROS. This model produced scores of 70.75%, 86.22%, and 72.41% for precision, recall, and F-measure, respectively. The use of only one classifier is a limitation of the study. Another shortcoming is the lack of information on the data cleaning process for the various datasets used.

[1]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[2]https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT
[3]https://scikit-learn.org/stable/
[4]https://spark.apache.org/mllib/

Churcher et al., 2021 [8] (An Experimental Analysis of Attack Classification Using Machine Learning in IoT Networks)

With regard to binary and multi-class classification, Bot-IoT was used in the evaluation of seven machine learning models: *k-Nearest Neighbor* (k-NN) [19], *support vector machine* (SVM) [20], decision tree [6], Naive Bayes [6], *random forest* (RF) [21], feedforward neural network, and *logistic regression* (LR) [6]. Models were implemented with Keras [5] and Scikit-learn. A sample size of about 1,500,000 instances was used, and an 80:20 split generated a training set and test set after data preprocessing, feature extraction and feature scaling. To address class imbalance, class weights were used in the training set. This involved dividing the number of dataset instances by the product of the number of classes in the dataset and the number labels. For binary classification, RF outperformed the other classifiers, with top scores of accuracy, precision, recall, and F-measure of 100% and an AUC of 1. We note that a complete description of a k-NN classifier for any study should include the value of k. As the value of k has not been provided, we believe this is a shortcoming of the paper.

Demirpolat et al., 2020 [9] (ProtÉdge: A Few-Shot Ensemble Learning Approach to Software-Defined Networking-Assisted Edge Security)

ProtÉdge, an ensemble of prototypical networks [22] and SVM, was evaluated against four other models: *convolutional neural network* (CNN) [23], SVM, deep autoencoders [24], and Naive Bayes. The proposed ensemble uses few-shot learning [25] to overcome the problem of training machine learning models with limited data. All models were trained on three different datasets: Bot-IoT, UNSW-NB15, and a software-defined networking [26] customized set. Models were built with Keras, Pytorch [6], and Scikit-learn. To deal with class imbalance, the number of instances in each category of Bot-IoT, except the Normal and Theft categories, was down-sampled to 20,000 during preprocessing. Instances for Normal and Theft were left untouched due to their comparatively small number. Normalization was then applied. For the training set, the authors randomly selected instances of 100, 400, 800, and 1,000. The authors also randomly selected 100 instances for the validation set, and the remaining instances from the down-sampled dataset as the test set. Results show that ProtÉdge outperformed the other models (in terms of accuracy, precision, recall, and F-measure) by about 20%. ProtÉdge obtained a best F-measure score of 96%. One limitation of this study is the small sizes of the training sets.

Ge et al., 2019 [10] (Deep Learning-based Intrusion Detection for IoT Networks)

A feedforward neural network and an SVM were trained on Bot-IoT to evaluate performance for binary and multi-class classification. Models were implemented with Scikit-learn,

[5]https://github.com/keras-team/keras
[6]https://pytorch.org/

TABLE II
BOT-IOT: PERFORMANCE SCORES

Author	Accuracy[1]	Precision[1]	Recall[1]	F-measure[1]	AUC
Bagui & Li, 2021 [7]	n/a	70.75	86.22	72.41	n/a
Churcher et al., 2021 [8]	100	100	100	100	1
Demirpolat et al., 2020 [9]	n/a	n/a	n/a	96.00	n/a
Ge et al., 2019 [10]	100	99.90	99.90	99.90	n/a
Ge et al., 2021 [11]	100	100	100	100	n/a
Koroniotis et al., 2020 [12]	99.90	100	99.90	99.90	n/a
Liaqat et al., 2020 [13]	99.99	99.83	99.33	99.33	n/a
Mulyanto et al., 2021 [14]	99.83	99.93	96.89	98.30	n/a
Soe et al., 2019 [15]	n/a	100	100	100	n/a

[1] All scores shown are percentages

TABLE III
BOT-IOT: TECHNIQUES FOR ADDRESSING CLASS IMBALANCE

Author	Technique
Bagui & Li, 2021 [7]	Random Oversampling
Churcher et al., 2021 [8]	Cost-Sensitive
Demirpolat et al., 2020 [9]	Random Undersampling
Ge et al., 2019 [10]	Cost-Sensitive
Ge et al., 2021 [11]	Cost-Sensitive
Koroniotis et al., 2020 [12]	Cost-Sensitive
Liaqat et al., 2020 [13]	Random Oversampling
Mulyanto et al., 2021 [14]	Cost-Sensitive
Soe et al., 2019 [15]	SMOTE

TABLE IV
BOT-IOT: PROPOSED MODELS

Author	Proposed Model
Bagui & Li, 2021 [7]	Feedforward Neural Network
Churcher et al., 2021 [8]	Random Forest
Demirpolat et al., 2020 [9]	ProtÉdge (Prototypical Networks + SVM)
Ge et al., 2019 [10]	Feedforward Neural Network
Ge et al., 2021 [11]	Feedforward Neural Network
Koroniotis et al., 2020 [12]	Particle Deep Framework (Deep Neural Network + Particle Swarm Optimization)
Liaqat et al., 2020 [13]	Convolutional Neural Network + Long Short-Term Memory
Mulyanto et al., 2021 [14]	Deep Neural Network
Soe et al., 2019 [15]	Feedforward Neural Network

TensorFlow [7], Keras, and Google Colab [8]. About 11,175,000 Bot-IoT instances were selected. After feature extraction and data preprocessing, the dataset was split in a 64:16:20 ratio for training, validation, and testing, respectively. To address class imbalance, higher weights were assigned to underrepresented classes. Class weights for the training data were obtained by dividing the packet count for each class by the total packet count and then inverting the quotient. Classification results show that the neural network outperformed the SVM. For binary classification, the best score for accuracy was 100%, while the best scores for precision, recall, and F-measure were all 99.90%. Unlike the multi-class model, the SVM classifier was not evaluated against the binary-class model. This is a limitation of the study.

[7]https://www.tensorflow.org/
[8]https://colab.research.google.com/

Ge et al., 2021 [11] (Towards a Deep Learning-Driven Intrusion Detection Approach for Internet of Things)

After selecting about 11,250,000 instances from Bot-IoT, the authors used transfer learning [27] to relay encoding between two feedforward neural networks (from a multi-class network to a binary-class network). The neural networks were built with TensorFlow and Keras. For evaluation purposes, an SVM model was built with Scikit-learn and also trained on Bot-IoT. Following feature extraction and data preprocessing, the dataset was split in a 64:16:20 ratio for training, validation, and testing, respectively. Class imbalance was tackled by giving higher weight values to underrepresented classes. Class

weights were obtained by dividing the packet count for each class by the total packet count and then inverting the quotient. For the binary-class network, several subcategories of attacks showed accuracy, precision, recall, and F-measure scores of 100%. Unlike the multi-class model, the SVM classifier was not evaluated against the binary-class model. This is a shortcoming of the paper.

Koroniotis et al., 2020 [12] (A New Network Forensic Framework Based on Deep Learning for Internet of Things Networks: A Particle Deep Framework)

A Particle Deep Framework is a proposed network forensics model that authenticates network data flows and uses a *deep neural network* (DNN) [28] based on *particle swarm optimization* (PSO) [29] to identify traffic anomalies. The framework was built with Keras, TensorFlow and Optunity [9], and trained on Bot-IoT. The authors used a sample size of 3,668,522 instances, which was split in an 80:20 ratio for training and testing. The logic cost function [30] was utilized, as it has been shown to be efficient at separating normal from attack traffic. The cost function is defined by the following equation [12]:

$$-\frac{1}{m}\sum_{i=1}^{m}(y_i log(\hat{y_i}) + (1-y_i) log(1-\hat{y_i})) \quad (1)$$

To treat class imbalance, weights for normal traffic w_0 and attack traffic w_1 were incorporated into this cost function. This modified equation is defined as follows: equation [12]:

$$-\frac{1}{m}\sum_{i=1}^{m}(w_1 y_i log(\hat{y_i}) + w_0(1-y_i) log(1-\hat{y_i})) \quad (2)$$

The best scores obtained by the model for accuracy, precision, recall, and F-measure were 99.90%, 100%, 99.90%, and 99.90%, respectively. Model performance was evaluated against the reported performance of models from other studies. The authors state that their model outperformed these other models (Naive Bayes, *multilayer perceptron* (MLP) [31], *association rule mining* (ARM) [32], decision tree, SVM, *recurrent neural network* (RNN) [33], and *long short-term memory* (LSTM) [34]). Evaluating model performance from one study against reported model performance from a non-identical study is problematic. This is the main shortcoming of the paper.

Liaqat et al., 2020 [13] (SDN Orchestration to Combat Evolving Cyber threats in Internet of Medical Things (IoMT))

Using a hybrid of two neural networks, CNN and *CUDA deep neural network LSTM* (cuDNNLSTM) [10], the authors set out to prove that their proposed model could outperform a *deep neural network-gated recurrent unit* (DNN-GRU) [35] hybrid, as well as a *long short-term memory-gated recurrent unit* (LSTM-GRU) [35] hybrid. The dataset sample contained 477 normal instances and 668,522 attack instances from Bot-IoT. Models were implemented with Keras and TensorFlow. During data preprocessing, the data was normalized, feature extraction was performed, and to address class imbalance, the number of normal instances was up-sampled to 2,400. The hybrid CNN-cuDNNLSTM model was shown to be the best performer with top scores of 99.99%, 99.83%, 99.33%, and 99.33% for accuracy, precision, recall and F-measure, respectively. The authors do not provide a reason for up-sampling the number of normal instances to 2,400, as opposed to a higher or lower value. This is a shortcoming of the research.

Mulyanto et al., 2021 [14] (Effectiveness of Focal Loss for Minority Classification in Network Intrusion Detection Systems)

To deal with class imbalance, the authors proposed a cost-sensitive neural network based on focal loss [36]. The cross-entropy loss function [37], which is widely used in neural network classification models, is integrated with focal loss to reduce the influence of the majority class(es) for binary and multi-class classification. A CNN and a DNN served as the neural network classifiers for this approach. The networks were trained on the NSL-KDD [38], UNSW-NB15, and Bot-IoT datasets. The Bot-IoT dataset sample contained about 3,000,000 instances. Models were constructed with Keras and TensorFlow. For binary classification, the cost-sensitive neural networks based on focal loss outperformed neural networks where SMOTE was applied and also outperformed plain neural networks. For Bot-IoT, top scores were obtained for the DNN cost-sensitive, focal-loss model: 99.83% (accuracy), 99.93% (precision), 96.89% (recall), and 98.30 (F-measure). One limitation of this paper is the inadequate amount of information provided on the preprocessing stage for Bot-IoT.

Soe et al., 2019 [15] (DDoS Attack Detection Based on Simple ANN with SMOTE for IoT Environment)

With a focus on DDoS attacks, the authors trained a feedforward neural network on 477 normal instances and about 1,900,000 DDoS instances from Bot-IoT. The neural network was implemented with Scikit-learn and Imblearn [11]. The dataset was split in a ratio of 66:34 for training and testing, and the SMOTE algorithm was introduced to address class imbalance. After the application of SMOTE during data preprocessing, the training dataset contained about 1,300,000 instances for each class, while the test dataset contained 655,809 normal instances and 654,285 DDoS instances. The data was then normalized. Precision, recall, and F-measure scores were all 100%. One limitation of this study is the lack of information on data cleaning. In addition, it is unclear why the authors believe that balancing the classes (positive to negative class ratio of 50:50) is the optimal solution. A third limitation is the reliance on only one classifier for performance results.

III. Discussion of Surveyed Works

In eight out of the nine works, the best performance scores for Bot-IoT are noticeably high. Accuracy scores are above 96%, and several precision, recall, and F-measure scores are

[9]https://optunity.readthedocs.io/en/latest/

[10]https://developer.nvidia.com/cudnn

[11]https://imbalanced-learn.org/stable/

100%. Furthermore, the only work that uses AUC showed a perfect score. The consistent reports of high scores among most of the works may indicate that Bot-IoT has few deficiencies and is relatively easy to classify. By contrast, a dataset such as KDD99 has known issues, including redundant and duplicate instances [39], thus making classification results inconsistent across diverse learners. Among the surveyed works, the paper by Bagui & Li, 2021 [7] is an anomaly, with precision, recall, and F-measure scores of 70.75%, 86.22%, and 72.41%, respectively. These comparatively lower scores may be due to the evaluation of only one classifier in the study. We point out that the data cleaning process could also be responsible, since the authors have not provided information on this procedure.

About 60% of data scientists are convinced that data cleaning is the most time-consuming task associated with machine learning [40]. We note that for three out of the nine surveyed works, the authors have not discussed their data cleaning process. A discussion of this procedure in any research paper should provide in-depth information on all instances and features of a dataset that have been deleted or modified. Inadequate or missing information on data cleaning can make duplication of an experiment difficult for outside researchers. If data cleaning has not been performed on a dataset such as Bot-IoT, this could result in inaccurate data analytics.

To address class imbalance, the authors adopted either a data-level approach or an algorithm-level approach. There was no apparent preference for one over the other. For the data-level approach, the authors obtained best results with either RUS, ROS, or SMOTE. For the algorithm-level approach, the authors used only cost-sensitive techniques. As the number of works addressing Bot-IoT class imbalance increases in the future, it is possible that there may be an observable trend toward a particular technique.

Interestingly, the accuracy metric is used for several surveyed works, while the AUC metric is only used in one study (Churcher et al., 2021). We note that relying solely on the accuracy metric in a study may not be advisable, since a deceptively high score could be obtained if the minority class is greatly underrepresented. It is more practical to provide accuracy along with other metrics, such as F-measure, and the authors of all the surveyed works should be commended for doing this. Furthermore, we recommend using the AUC metric, which is robust to class imbalance.

Based on our experimentation, we discovered there are six Bot-IoT features that inflate performance scores. They are as follows: *pkSeqID*, *seq*, *stime*, *ltime*, *saddr*, and *daddr*. We consider them to be invalid for building predictive models in the intrusion detection domain. Among the surveyed works, only Demirpolat et al. [9] clearly show that their feature set is free of these six attributes.

Finally, the results of our survey indicate that statistical analysis of performance scores has been ignored across-the-board. Assessing the statistical significance of such scores increases clarity, and there are some standard tests for doing this, including *ANalysis Of VAriance* (ANOVA) [41] and Tukey's *Honestly Significant Difference* (HSD) [42]. ANOVA shows whether the means of one or more independent factors are statistically significant. Tukey's HSD assigns letters to group means that are significantly different from each other.

A. Gaps in Current Research

There are gaps in the research associated with tackling class imbalance in Bot-IoT. Specifics such as concept drift and the use of feature selection to mitigate class imbalance are missing from the literature. We expand on these topics in the following paragraphs.

Concept drift refers to the change in data distributions over time [43]. For example, a model trained in 2021 on Bot-IoT may have a lower F-measure score in 2030, when evaluated against a modern intrusion detection dataset. Some attack instances from Bot-IoT would most likely be ineffective in the future (updated software, patches etc.). Research investigating the influence of time on intrusion detection models is a blossoming area.

As stated earlier, the use of feature selection to address class imbalance is a data-level approach. Feature selection algorithms, such as RELIEFF, can distribute higher weights to attributes linked to a minority class [44], and also improve classification performance while reducing computational cost. Within the context of class imbalance, a feature selection algorithm may be suitable for use with one learner but not with another.

IV. Conclusion

Society cannot afford to ignore the increasing security risk associated with the explosive growth in the number of IoT devices. Bot-IoT, an IoT-centric dataset for network intrusion detection, is a tool for confronting this risk. The dataset is class-imbalanced and contains about 73,000,000 instances. Our survey examines Bot-IoT research papers that use techniques for addressing class imbalance.

In general, we noticed that performance scores were exceptional. This finding points to the high reliability of Bot-IoT, in addition to hinting that Bot-IoT is relatively easy to classify. Secondly, we discovered that information on data cleaning was excluded in a few papers. Finally, we noted the popularity of both data-level and algorithm-level approaches for mitigating class imbalance in Bot-IoT.

A few gaps have been identified within the surveyed works. Particulars such as concept drift and the use of feature selection techniques to address class imbalance are missing from the corpus. Future work should address these gaps.

References

[1] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[2] Microsoft, "Microsoft report shows increasing sophistication of cyber threats," https://blogs.microsoft.com/on-the-issues/2020/09/29/microsoft-digital-defense-report-cyber-threats/.

[3] Forbes, "5 iot trends to watch in 2021," https://www.forbes.com/sites/danielnewman/2020/11/25/5-iot-trends-to-watch-in-2021/?sh=4643ae1201b3.

[4] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[5] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *Journal of Big Data*, vol. 5, no. 1, p. 42, 2018.

[6] R. Zuech, J. Hancock, and T. M. Khoshgoftaar, "Detecting web attacks using random undersampling and ensemble learners," *Journal of Big Data*, vol. 8, no. 1, pp. 1–20, 2021.

[7] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *Journal of Big Data*, vol. 8, no. 1, pp. 1–41, 2021.

[8] 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.

[9] A. Demirpolat, A. K. Sarica, and P. Angin, "Protédge: A few-shot ensemble learning approach to software-defined networking-assisted edge security," *Transactions on Emerging Telecommunications Technologies*, p. e4138, 2020.

[10] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep learning-based intrusion detection for iot networks," in *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE, 2019, pp. 256–25 609.

[11] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for internet of things," *Computer Networks*, vol. 186, p. 107784, 2021.

[12] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for internet of things networks: A particle deep framework," *Future Generation Computer Systems*, vol. 110, pp. 91–106, 2020.

[13] S. Liaqat, A. Akhunzada, F. S. Shaikh, A. Giannetsos, and M. A. Jan, "Sdn orchestration to combat evolving cyber threats in internet of medical things (iomt)," *Computer Communications*, vol. 160, pp. 697–705, 2020.

[14] M. Mulyanto, M. Faisal, S. W. Prakosa, and J.-S. Leu, "Effectiveness of focal loss for minority classification in network intrusion detection systems," *Symmetry*, vol. 13, no. 1, p. 4, 2021.

[15] Y. N. Soe, P. I. Santosa, and R. Hartanto, "Ddos attack detection based on simple ann with smote for iot environment," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*. IEEE, 2019, pp. 1–5.

[16] H. Shamsudin, U. K. Yusof, A. Jayalakshmi, and M. N. A. Khalid, "Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset," in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*. IEEE, 2020, pp. 803–808.

[17] S. Varsamopoulos, B. Criger, and K. Bertels, "Decoding small surface codes with feedforward neural networks," *Quantum Science and Technology*, vol. 3, no. 1, p. 015004, 2017.

[18] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[19] S. Vajda and K. Santosh, "A fast k-nearest neighbor classifier using unsupervised clustering," in *International conference on recent trends in image processing and pattern recognition*. Springer, 2016, pp. 185–193.

[20] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE access*, vol. 6, pp. 33 789–33 795, 2018.

[21] T. M. Khoshgoftaar, M. Golawala, and J. Van Hulse, "An empirical study of learning from imbalanced data using random forest," in *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, vol. 2. IEEE, 2007, pp. 310–317.

[22] T. Gao, X. Han, Z. Liu, and M. Sun, "Hybrid attention-based prototypical networks for noisy few-shot relation classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6407–6414.

[23] G. Kaur, A. H. Lashkari, and A. Rahali, "Intrusion traffic detection and characterization using deep image learning," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, 2020, pp. 55–62.

[24] Y. Pan, F. He, and H. Yu, "Learning social representations with deep autoencoder for recommender system," *World Wide Web*, vol. 23, no. 4, pp. 2259–2279, 2020.

[25] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.

[26] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[27] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[28] G. C. Amaizu, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Investigating network intrusion detection datasets using machine learning," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2020, pp. 1325–1328.

[29] A. J. Malik and F. A. Khan, "A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection," *Cluster Computing*, vol. 21, no. 1, pp. 667–680, 2018.

[30] M. De Cock, R. Dowsley, A. C. Nascimento, D. Railsback, J. Shen, and A. Todoki, "High performance logistic regression for privacy-preserving genome analysis," *BMC Medical Genomics*, vol. 14, no. 1, pp. 1–18, 2021.

[31] T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Supervised neural network modeling: an empirical investigation into learning from imbalanced data with labeling errors," *IEEE Transactions on Neural Networks*, vol. 21, no. 5, pp. 813–830, 2010.

[32] G. Ceddia, L. N. Martino, A. Parodi, P. Secchi, S. Campaner, and M. Masseroli, "Association rule mining to identify transcription factor interactions in genomic regions," *Bioinformatics*, vol. 36, no. 4, pp. 1007–1013, 2020.

[33] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[34] P. Lin, K. Ye, and C.-Z. Xu, "Dynamic network anomaly detection system by using deep learning techniques," in *International Conference on Cloud Computing*. Springer, 2019, pp. 161–176.

[35] S. Nakayama and S. Arai, "Dnn-lstm-crf model for automatic audio chord recognition," in *Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence*, 2018, pp. 82–88.

[36] K. Nemoto, R. Hamaguchi, T. Imaizumi, and S. Hikosaka, "Classification of rare building change using cnn with multi-class focal loss," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 4663–4666.

[37] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2019.

[38] L. Dhanabal and S. Shantharajah, "A study on nsl-kdd dataset for intrusion detection system based on classification algorithms," *International journal of advanced research in computer and communication engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[39] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. IEEE, 2009, pp. 1–6.

[40] Z. Groff and S. Schwartz, "Data preprocessing and feature selection for an intrusion detection system dataset," in *34th Annual Conference of The Pennsylvania Association of Computer and Information Science Educators*, 2019, pp. 103–110.

[41] G. R. Iversen, A. R. Wildt, H. Norpoth, and H. P. Norpoth, *Analysis of variance*. Sage, (1987).

[42] J. W. Tukey, "Comparing individual means in the analysis of variance," *Biometrics*, pp. 99–114, 1949.

[43] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data," *Journal of Big Data*, vol. 7, no. 1, pp. 1–19, 2020.

[44] D. Tiwari, "Handling class imbalance problem using feature selection," *International Journal of Advanced Research in Computer Science & Technology*, vol. 2, no. 2, pp. 516–520, 2014.