Towards Effective Network Intrusion Detection in Imbalanced Datasets: A Hierarchical Approach

Md. Shamim Towhid, Nasik Sami Khan, Md Mahibul Hasan, Nashid Shahriar

Department of Computer Science, University of Regina
{mty754, nku618, mhr993 nashid.shahriar}@uregina.ca

***Abstract*—The transition from conventional networks to Software Defined Networks (SDNs) has revolutionized network management and control, but it also creates a huge security risk, underscoring the significance of effective intrusion detection systems (IDS). Researchers have used deep learning for IDS due to its ability to capture complex patterns in data. Deep Learning techniques rely on ample balanced labeled data for effective intrusion detection, but acquiring such balanced data in real network scenarios is a formidable challenge, often resulting in suboptimal performance for existing methods when dealing with imbalanced datasets. This paper introduces a hierarchical approach that is capable of effectively detecting well-known network attacks in an SDN environment, even with minimal training data. Our model, leveraging a dataset collected from a real-world software-defined wide area network (SD-WAN) environment, showcases remarkable adaptability by maintaining strong performance even with highly imbalanced data, i.e., attack samples with as few as 8 or 16 instances to others with hundreds, thousands, or even millions of instances. It consistently achieves an overall average F1 score above 92%, with minority class average F1 score reaching more than 84%, marking a substantial 22.50% performance improvement compared to selected baselines in our evaluation.**

***Index Terms*—intrusion detection, deep learning, flow-based classification, SDN**

I. Introduction

With the advent of Software Defined Networks (SDNs) in recent years, the networking landscape has undergone a profound transformation. This paradigm shift has ushered in unprecedented flexibility and efficiency in network management. While this architectural change offers numerous benefits, it has also introduced a significant security concern, the potential vulnerability of the centralized controller, which, if compromised, could have catastrophic repercussions for the entire network. As attacks continue to evolve at a rapid pace, each requiring distinct treatment, the imperative arises to detect all potential attack classes within network traffic. The research trend on intrusion detection is going towards implementing deep learning models [14]. However, the scarcity of sufficient training samples from all attack classes causes the deep learning models to perform poorly on these attack classes. Therefore, some researchers [2], [4] have implemented a grouping strategy to solve this problem. This grouping is done by combining multiple classes depending on their characteristics.

Deep learning models require a balance between different types of class in the training dataset. Existing works have shown the disadvantage of having class imbalance in the training dataset [2], [4], [9]. The data imbalance specifically causes performance degradation for deep learning models compared to the traditional machine learning models because the gradient update is mostly done by the majority classes. This introduces a bias towards these majority classes during the training. One common way to solve this issue is to use augmentation to upsample the minority classes which is adapted by existing works from the literature. However, there is a limitation in this approach. We can upsample the minority classes up to a certain limit. If the dataset is highly imbalanced then the augmentation processes can cause overfitting of the model.

This paper is targeted at addressing the challenges faced by extremely uneven attack samples. We utilize a unique dataset obtained from an actual SD-WAN network, encompassing 15 different classes. We intensively study several strategies and finally propose a hierarchical learning approach that surpasses the compared approaches from literature, both on an individual class basis and holistically. Our approach is evaluated on two different datasets (described in Section III-A).

Our main contributions in this paper are as follows:

- We propose a hierarchical approach that combines deep learning models with a traditional machine learning model to address the data imbalance issue across attack classes.
- Several experiments are done using two available datasets to demonstrate the effectiveness of the proposed approach. We show that the proposed approach outperforms the best-performing compared approach by 22.50% on the minority classes in terms of F1 score.
- We showcase the generalization ability of the proposed approach by training it in one dataset and testing the trained model on another dataset that has similar characteristics. This experiment shows the performance of the models does not decrease on the new test dataset.

The remainder of the paper is structured as follows: in Section II, we examine previous research on IDS. Section III describes the methodology of our approach in addition to the description of the used datasets and our data preparation steps, while Section IV evaluates the results. Section V concludes the paper and discusses prospective research.

II. Related Works

In this section, we explore state-of-the-art research focusing on detecting intrusions in the SDN environment. In SDN networks, there is a dearth of appropriate datasets for intrusion detection. Since the performance of the intrusion detection system extensively relies on the characteristics of the dataset, SDN-specific dataset that contains real network traffic is necessary.

As a solution to the shortage of specialized datasets in SDN configurations, the authors in [1] developed one of the first comprehensive SDN datasets named InSDN, tailored for evaluating IDS performance. The dataset comprises seven major categories of attacks, including DoS attacks, Malware, Web Attacks, etc. The study evaluated the effectiveness of several ML models and found that the Random Forest method outperforms other models.

The study in [2], [3], and [4] used the dataset published by [1]. The effectiveness of the XGBoost algorithm in detecting multiple classes of attacks and the significance of dataset selection is emphasized by [2]. The study examines machine learning models' performance in SDN settings and finds high accuracy in classifying attacks from the same source, but a noticeable 28% fall when evaluating data from different sources.

The study in [3] presents a hybrid approach to intrusion detection by employing traditional ML algorithms and deep learning methods. The authors introduce a new regularization function with a hybrid model to handle overfitting, making it effective in both binary and multi-class classification. The method extracts features from training data and uses them as input for machine learning models. The regularization outperforms L1 and L2 regularization.

The study in [4] presents a novel method for intrusion detection by framing it as an anomaly detection task. The authors integrate an LSTM autoencoder with the One-class SVM (OC-SVM) method using a hybrid approach. The autoencoder, trained in two steps, detects data patterns and abnormalities and then uses compressed representations for binary classification in OC-SVM. However, the accuracy score is stated to be lower than other results addressed in [2] and [3].

The InSDN dataset was developed in the SDN Network environment. However, the publishers of [5] introduced a new dataset with the same number of features as the InSDN dataset, named CICIDS 2017, which was developed in a conventional network environment. There are thirteen classes, which include various attack sample types. As the previously discussed dataset, this also has attack class imbalance issue. To assess the efficiency of the features in identifying the specified attack families, the study in [5] examines the performance of the selected extracted features using seven widely used ML techniques. The authors in [6], [7], and [8] used the CICIDS2017 dataset to develop their solutions. The study [6] introduces a multi-stage approach for hierarchical intrusion detection. It focuses on a multi-stage detection mechanism. Initially, it detects anomalies using an Autoencoder and then integrates this information with a One-Class Support Vector Machine (OC-SVM). Subsequently, a Random Forest (RF) and Neural Network (NN)-based multi-class classification method is applied. Finally, the overall performance of the three classification steps is assessed. The authors combined multiple categories of attacks into sub-categories, based on their characteristics. This grouping helped to reduce class diversity and contributed to solving data imbalance issues.

The authors in [8] introduce the implementation of Deep Belief Networks (DBNs) for Network Intrusion Detection Systems (NIDS) using the CIC-IDS-2017 dataset. Through a two-stage training procedure involving unsupervised pre-training and supervised fine-tuning, the DBNs provide a unique approach to collecting high-dimensional representations. The authors explore and evaluate various class-balancing techniques to address the challenge of the imbalanced dataset. The proposed combination of Synthetic Minority Over-sampling Technique (SMOTE) [9] and random undersampling is identified as the highest effective method, significantly enhancing the detection performance. In this paper we present a hierarchical approach combining deep learning and traditional ML models, demonstrating its effectiveness and generalization ability through training on one dataset, and testing on a separate dataset.

III. Methodology

We describe our proposed hierarchical approach to deal with highly imbalanced datasets for intrusion detection in this section. First, we describe the used datasets in our experiments followed by the feature analysis and data pre-processing steps. Second, we explain our proposed approach in detail. Finally, we conclude this section by discussing the compared approaches in our experiments.

A. Dataset Preparation

ITU Challenge Dataset: The ITU challenge dataset is provided by the ULAK communication as a part of an ITU challenge [10] where the goal is to develop a machine learning model to classify network flows into benign and multiple types of attack flows. Some network flows of this dataset are collected from real users in an SD-WAN environment prepared by the ULAK communication. These real flows are combined with some known datasets from the literature. There are in total 15 classes in this dataset that fall under four primary categories: Benign, DDoS, Malware, and Web-based attack. The number of samples per class in this dataset is shown in Table I. From Table I it is clear that the dataset is highly skewed which is the main challenge to solve in this dataset.

CICIDS2017: The CICIDS2017 dataset is a well-known dataset in the literature which is made available by the Canadian Institute for Cybersecurity [11]. The abstract behavior of human interactions is modeled using a proposed system [11] and then the benign network flows are collected by simulating 25 users behavior in a network. The attackers are simulated from outside of the victim network to collect attack traffic. The victim network includes Modems, Firewalls, Switches, Routers, and a variety of operating systems such as Windows, Ubuntu, and Mac OS X. The number of classes and features are exactly the same as the ITU challenge dataset.

We select these two datasets to showcase the effectiveness of our proposed method in handling highly skewed data. Table I shows the number of flows per class in the CICIDS2017 dataset and it is clear that this dataset has the same problem of imbalanced data as the ITU challenge dataset.

TABLE I
NUMBER OF SAMPLES PER CLASS IN THE SELECTED DATASETS

Class Names	ITU Challenge	CICIDS2017	Group
Benign	1,842,915	2,271,320	Large
DoS_Hulk	187,201	230,124	Large
PortScan	128,853	158,804	Large
DDoS	103,816	128,025	Large
DoS_Golden_Eye	8,345	10,293	Large
FTP-Patator	6,436	7,935	Large
SSH-Patator	4,781	5,897	Large
DoS_sloworis	4,699	5,796	Large
DoS_slowhttptest	4,458	5,499	Large
Bot	1,592	1,956	Small
Brute-Force	1,221	1,507	Small
XSS	527	652	Small
Infiltration	28	36	Small
SQL_Injection	16	21	Small
Heartbleed	8	11	Small

B. Data Preparation

Since deep learning models are used in our proposed approach we use a standard scaler from scikit-learn to scale the data by removing the mean and scaling to unit variance. This scaling is necessary because large input values can lead to numerical instability during training. We use the numerical augmentation technique SMOTE [9] by following [3] to increase the number of samples in some of the minority classes. The number of samples for minority classes after augmentation is given in Table II. The augmented data are used only for training. The testing is done on original data from the dataset.

TABLE II
NUMBER OF SAMPLES PER CLASS IN ITU CHALLENGE DATASET AFTER AUGMENTATION

Class Name	New Number of samples
Bot	7232
Brute-Force	949
XSS	820
Infiltration	422
SQL_Injection	412
Heartbleed	406

We identify 28 important features from the dataset by using the feature importance score of the RF model. RF assigns feature importance scores by considering how each feature contributes to reducing error in the decision trees during training. Combining these scores from all the trees, the RF assigns a feature importance score to each feature. Features that consistently help in making accurate predictions across the ensemble are considered more important than others. We train an RF model on the dataset using all the features and calculate this feature importance score.

C. Our Hierarchical Approach

By observing the number of instances per class and the nature of the attack traffic flows, we divide the dataset into two groups. One group contains the classes with a large number of samples in our dataset. Another group contains classes with the smaller number of samples. All the DoS and DDoS attacks including the SSH-Patator, FTP-Patator, and PortScan attacks fall under the large group based on the number of samples. The benign class belongs to the large group as well. The small group consists of all the web-based attacks, infiltration, Bot, and Heartbleed attacks. The benefit of categorizing in this way is to use two separate machine learning models on these two groups to achieve high performance in each individual class. We can combine the prediction from these two separate models by using another model that will classify a data sample into either large or small groups.

This idea of hierarchical classification is inspired by the fact that traditional machine learning approaches like RF and Xgboost are able to achieve high performance when less training data are available. On the other hand, deep learning models like Convolutional Neural Network (CNN) requires a large number of labeled data. CNNs are capable of automatically learning relevant features from raw data which is beneficial in our case because we are dealing with a large number of features in both datasets. Furthermore, CNNs are better known for their transferability in the literature which is crucial in intrusion detection because we can fine-tune the learned parameters of CNN from one dataset to new types of attacks. On the other hand, the ensemble nature of RF results in models with lower bias and variance, leading to robust performance. Therefore, we train a CNN-based model on the large group and an RF model on the small group. In our experiment, we use a smaller version of ResNet [13] that has four residual blocks to train on the large group. An RF model with default parameters from the scikit-learn library is used to train on data from the small group. Another CNN model with the same architecture is used for training to classify each sample to either large or small groups. In our experiment, the CNN model is trained with all the features, and the RF model is trained with the selected 28 features to get the best performance.

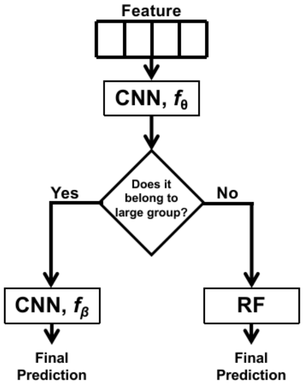

Fig. 1. The inference process of the hierarchical approach

Figure 1 shows the inference process of the proposed method. At first, the features are given to the first CNN model (f_θ) where the data sample is classified into two classes: Large and Small. Then based on the prediction of the first CNN model, either the second CNN model (f_β) or the RF model is used for the final classification. This hierarchical classification allows us to deal with the class imbalance in the dataset and the proposed approach achieves high accuracy, precision, and recall score in each individual class.

We train both the CNN models (f_θ and f_β) for 100 epochs by enabling the early stopping technique with a patience value of 12. The batch size is set to 512. We use a single 11^{th} Gen i7 machine with one GeForce RTX 3090 GPU with 24GB memory and 32GB main memory for the training. The Adam optimizer is used for updating the weights of the CNN models. Each experiment is run three times to get a statistically stable result. Each time we change the random state variable for the RF model and XGBoost model.

D. *Compared Approaches*

There are a plethora of studies on network-based intrusion detection systems using machine learning in the literature. We found the approach mentioned in [3] interesting as the authors propose a hybrid model that combines CNN and RF for intrusion detection in an SDN environment. The dataset used in this work has similar imbalanced class distributions. Hence, we select this approach as one of our compared approaches. We implement the approach mentioned in [3] and use that as a baseline to benchmark our approach in the evaluation part. Furthermore, we train RF, XGBoost, and CNN models without any hierarchical approach to compare the performance with our proposed approach. The results of each approach are described in detail in the following section.

IV. Evaluation

This section explains all our experiments and the obtained results in detail. In our first experiment, we try to compare our proposed approach with baseline methods described in Section III-D. As evaluation metrics, we select the most commonly used metrics to evaluate multi-class classification models from the literature. These are Accuracy (ACC), Precision (Pr), Recall (Rec), and F1 score. Since we are dealing with a highly imbalanced dataset, it is crucial to measure these evaluation metrics per class in our dataset.

We train each model three times and the average results on the test dataset of the ITU challenge are shown in Table III. The ITU Challenge dataset is divided into training (70%) and test (30%) set by the competition organizers. Table III shows the proposed approach outperforms all the baseline models in terms of precision, recall, and F1 scores. Although the accuracy of the XGBoost model is slightly better than the hierarchical approach, the recall score of XGBoost is significantly less than the hierarchical approach. The recall score is an indicator of the true positive rate of the model. A low recall score in a multi-class classification setting means that the model is not correctly classifying all the positive instances (the instances of the class we are interested in) in the dataset. The proposed hierarchical approach achieves a balanced result in terms of the precision and recall score. The overall improvement is realized by the F1 score because it is the harmonic mean of precision and recall score. A high F1 score means that the model does well in both precision and recall. The proposed hierarchical approach outperforms all the compared approaches in terms of F1 score by at least 8%.

TABLE III
Comparison of the proposed approach with selected baseline models

Model	**ACC (%)**	**Pr (%)**	**Rec (%)**	**F1 (%)**	**Inference time (sec)**
Hybrid [3]	99.85	72.33	72.53	72.41	3.65e-5
RF	99.84	88.63	79.64	81.35	1.40e-5
XGBoost	**99.87**	93.23	82.45	84.27	**2.07e-6**
CNN	99.41	89.00	67.81	71.51	8.08e-5
Hierarchical	99.02	**93.34**	**91.16**	**92.05**	1.61e-4

The inference time is a crucial factor especially for intrusion detection because it is the time taken by the model to classify one single network flow. Table III shows the inference time in seconds for each model. XGBoost is the fastest model in terms of inference time. However, the F1 score is not great compared to the hierarchical approach. Since the hierarchical approach combines both CNN and RF models the inference time is slightly higher than the compared approaches, remaining in a negligible range.

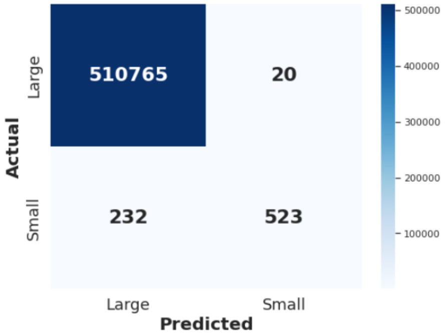

Fig. 2. Confusion matrix of the first CNN model

In our proposed approach, the first CNN model (f_θ) is used to classify whether the data sample belongs to the large group or the small group. A misclassification in this model would definitely misclassify the data samples. Therefore, it is essential to evaluate the performance of this model. Figure 2 shows the confusion matrix of this model on the test set of the ITU challenge dataset. From this confusion matrix, we can see the model misclassified 20 samples from the large category and 232 samples from the small category. Further hyperparameter tuning can improve this result. However, our goal is to achieve a balanced result per class which is obtained in spite of this misclassification by the first CNN model.

The class-wise F1 score on minority classes of all the models is shown in a grouped bar chart in Figure 3. We observe that some of the models perform poorly in the classes where we have a small number of training samples. For example, the F1 score of the hybrid model on class “Infiltration” is 0%. Hence, we do not see any portion of it in Figure 3. The

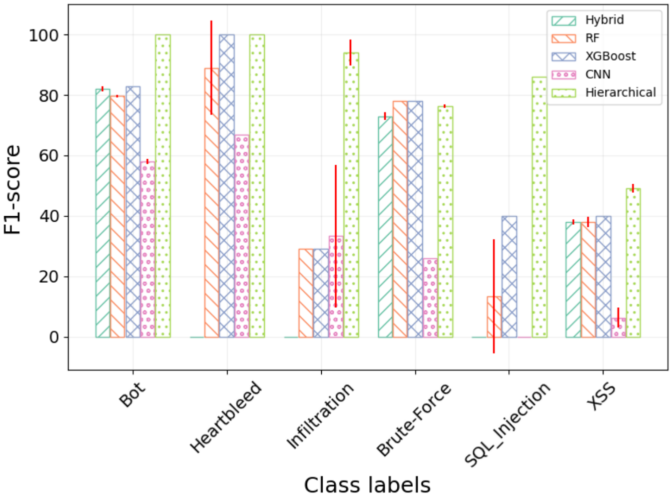

Fig. 3. Class-wise F1 score on the minority classes of ITU Challenge Dataset

error bar shows the standard deviation of the F1 score among the three runs for each experiment. The proposed hierarchical approach outperforms the second-best model (XGBoost) by 22.50% on average for the minority classes in terms of F1 score.

In our next experiment, we try to observe the generalization ability of the model that is trained with our proposed approach. For that, we test the model on the CICIDS2017 dataset without any additional training. In other words, the model is trained on the ITU challenge dataset and tested on the CICIDS2017 dataset. Table I shows that the total number of instances in the CICIDS2017 dataset is larger than the ITU challenge dataset. The class-wise result of this experiment is shown in Table IV. From this result, we observe that the overall class-wise performance of the model on the CICIDS2017 dataset is even better than the ITU challenge dataset. For example, the F1 score on the "XSS" class of the ITU challenge dataset is 51% whereas it is 79% in the CICIDS2017 dataset. This behavior demonstrates the generalization ability of the proposed approach.

TABLE IV

CLASS-WISE EVALUATION METRICS OF THE CICIDS2017 DATASET

Class Name	Pr (%)	Rec (%)	F1 (%)
Benign	99.97	99.66	99.81
Bot	99.95	100	99.97
DDoS	99.96	99.94	99.95
DoS_Golden_Eye	99.48	98.31	98.89
DoS_Hulk	97.40	99.84	98.60
DoS_slowhttptest	90.13	98.47	94.12
DoS_slowloris	98.90	98.86	98.88
FTP-Patator	99.72	99.68	99.70
Heartbleed	100	100	100
Infiltration	100	97.22	98.59
PortScan	99.36	99.94	99.65
SSH-Patator	98.07	98.02	98.04
Brute-Force	91.82	88.65	90.21
SQL_Injection	95.00	90.48	92.68
XSS	75.74	81.90	78.70

V. CONCLUSION

In this paper, we address the class imbalance problem for network intrusion detection using a hierarchical approach that combines deep learning and traditional machine learning models. Our experiments show the proposed method can outperform the compared approaches, especially in the minority classes. On average, the proposed approach outperforms the best-performing compared approach by 22.50% on the minority classes (classes belonging to the small group) in terms of F1 score. The proposed approach also outperforms the baselines in terms of precision, recall, and F1 score. Finally, we demonstrate the generalization ability of the proposed method by evaluating it on another reputed IDS dataset.

In future work, we want to explore the area of few-shot learning to address the data imbalance problem. Another interesting research direction would be to explore how we can leverage feature engineering with reinforcement learning to improve the model performance.

REFERENCES

[1] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," IEEE Access, vol. 8, pp. 165263–165284, 2020.

[2] Q.-V. Dang, "Intrusion detection in software-defined networks," presented at the Future Data and Security Engineering: 8th International Conference, FDSE 2021, Virtual Event, November 24–26, 2021, Proceedings 8, Springer, 2021, pp. 356–371.

[3] M. S. ElSayed, N.-A. Le-Khac, M. A. Albahar, and A. Jurcut, "A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique," Journal of Network and Computer Applications, vol. 191, p. 103160, 2021.

[4] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using LSTM based autoencoder," presented at the Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, 2020, pp. 37–45.

[5] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," ICISSp, vol. 1, pp. 108–116, 2018.

[6] M. Verkerken et al., "A Novel Multi-Stage Approach for Hierarchical Intrusion Detection," IEEE Transactions on Network and Service Management, 2023.

[7] M. N. Goryunov, A. G. Matskevich, and D. A. Rybolovlev, "Synthesis of a machine learning model for detecting computer attacks based on the cicids2017 dataset," Proceedings of the Institute for System Programming of the RAS, vol. 32, no. 5, pp. 81–94, 2020.

[8] O. Belarbi, A. Khan, P. Carnelli, and T. Spyridopoulos, "An intrusion detection system based on deep belief networks," presented at the International Conference on Science of Cyber Security, Springer, 2022, pp. 377–392.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.

[10] ULAK Communication. 2023. Intrusion and Vulnerability Detection in Software-Defined Networks. (May 2023). Retrieved October 16, 2023 from https://challenge.aiforgood.itu.int/match/matchitem/81

[11] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018.

[12] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[14] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., and Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150