

IntrusionHunter: Detection of Cyber Threats in Big Data

Hashem Mohamed^a, Alia El Bolock^b and Caroline Sabty^c

Informatics and Computer Science, German International University, Cairo, Egypt

Keywords: Intrusion Detection, Big data, Deep Learning, Machine Learning.

Abstract: The rise of cyber-attacks has become a serious problem due to our growing reliance on technology, making it essential for both individuals and businesses to use efficient cybersecurity solutions. This work continues on previous work to improve the accuracy of intrusion detection systems by employing advanced classification techniques and an up-to-date dataset. In this work, we propose IntrusionHunter, an anomaly-based intrusion detection system operating on the CSE-CICIDS2018 dataset. IntrusionHunter classifies intrusions based on three models, each catering to different purposes: binary classification (2C), multiclass classification with 7 classes (7C), and multiclass classification with 15 classes (15C). Four main classification models were used: Random Forest, Extreme Gradient Boosting, Convolutional Neural Networks, and Deep Neural Networks. The results show that Random Forest and XGBoost algorithms outperformed state-of-the-art intrusion detection systems in binary and multiclass classification (15 classes). The findings also show that the dataset imbalance needs to be addressed to improve the performance of deep learning techniques.

1 INTRODUCTION


The Internet and cloud computing have increased security breaches, with cyberattacks becoming more sophisticated and widespread (Burbank, 2008). The rise of big data poses a significant threat to cybersecurity, making it essential for organizations to implement robust security measures. Cybersecurity is complex and consists of multiple vital components in keeping hackers out and protecting sensitive information. Network security is essential for any modern organization, as it is the backbone of communication, collaboration, and productivity. However, network security faces the challenge of constantly evolving security threats, requiring a proactive and multi-layered approach to stay ahead of attackers (Dowd and McHenry, 1998). The concept of defense in depth is a fundamental principle of network security, where multiple layers of security are put in place to protect a network. Standard techniques and technologies used in network security include firewalls, encryption, authentication, and intrusion detection and prevention systems.


Intrusion Detection Systems (IDS) monitor network activity, alerting administrators to potential


threats before they cause significant damage and providing valuable insights into the nature and scope of security threats, helping organizations improve their overall security posture and prevent similar incidents. There are two main types of IDS: signature-based and anomaly-based. Signature-based IDS, also known as network-based IDS (NIDS), monitors network traffic and searches for known attack signatures, such as suspicious network traffic patterns and attempts to access prohibited network resources. Anomaly-based IDS monitors individual devices on a network, looking for patterns of behavior that deviate from regular activity and could indicate unauthorized access attempts (Marchang et al., 2017).

Previous studies in the field of intrusion detection systems have limitations, including the use of outdated datasets that may not accurately reflect current cybersecurity threats, and the focus on binary classification without considering different types of attacks. These limitations highlight the need for more recent and comprehensive studies that employ advanced classification techniques to improve the identification of different types of attacks.

This work presents an anomaly-based intrusion detection system using the CSE-CICIDS2018 dataset (Communications Security Establishment (CSE), 2018). The dataset underwent several preprocessing steps to ensure high-quality data for model train-

^a  <https://orcid.org/0009-0002-9003-699X>

^b  <https://orcid.org/0000-0002-5841-1692>

^c  <https://orcid.org/0000-0002-3590-5737>

ing, including feature selection to decrease training time. The system evaluates the performance of several popular machine learning algorithms, including Random Forest, XGBoost, Convolutional Neural Networks, and Deep Neural Networks, in binary and multiclass classifications. Results show that Random Forest and XGBoost algorithms outperform state-of-the-art intrusion detection systems in terms of accuracy and F1-score, and that handling the dataset imbalance can improve the performance of deep learning techniques.

The paper is structured as follows: Section 2 reviews previous related work on intrusion detection systems and the classification models used in the field. Section 3 presents the methodology used in this research, including the data preprocessing, feature selection, and evaluation of different classification models. Section 4 presents the evaluation results, discussing the performance of the different classification models for each type of classification. Finally, Section 5 provides conclusions and suggestions for future work.

2 RELATED WORK

This section summarizes the essential findings and contributions of previous studies, discussing their implications for the current research. The previous studies use various intrusion detection approaches, including machine learning and deep learning algorithms. It also identifies gaps in the existing literature and potential avenues for future research.

2.1 Machine Learning Approaches

Machine learning classification algorithms, including Random Decision Forest, Bayesian Network, Naive Bayes classifier, Decision Tree, Random Tree, Decision Table, and Artificial Neural Network, are used in cyber-security for intrusion detection as explored in (Alqahtani et al., 2020) using the KDD Cup 99 (Information and of California, 1999) dataset. IDS models based on Random Forest classifiers outperform other classifiers, particularly the Random Decision Forest, which has an accuracy of 99%, precision, recall of 93%, and F1-score of 0.97. The Random Forest model derives rules for the forest from a number of decision trees and generates more logic rules by considering the majority vote of these trees. Future work involves expanding cyber-security datasets and creating a data-driven intrusion detection system for automated security services.

The work presented in (He et al., 2019) looked

into the issue of machine learning-based threat detection in network security. The stochastic gradient descent enhanced the K-means clustering technique; therefore, the Support vector machine was coupled with it and was trained and tested on the KDD Cup 99 (Information and of California, 1999) dataset. It was discovered that the method used in this study had an (87.1%) detection rate and a (3.1%) false alarm rate, and that its detection effects were superior to those of both the SVM algorithm and a single K-means clustering algorithm. The dependability of the approach used in this investigation was demonstrated by the DoS detection rate (94.5%). The improved algorithm had a greater detection rate and a lower false alarm rate, showing that the upgrade to the clustering algorithm was successful.

2.2 Deep Learning Approaches

The performance of intrusion detection systems is enhanced by integrating big data and deep learning techniques in (Faker and Dogdu, 2019). Deep Feed-Forward Neural Networks (DNN) and two ensemble methods—Random Forest and Gradient Boosting Tree (GBT)—are used for classification. UNSW NB15 (Sydney,) and CICIDS2017 (for Cybersecurity (CIC), 2018) are the datasets used in this paper and include attacks several types of attacks. On the UNSW-NB15 dataset, the findings demonstrate very short prediction times and high accuracy levels with DNN for binary and multiclass classification (99.19 % and 97.04 %, respectively). Using the CICIDS2017 dataset, the GBT classifier had the best accuracy (99.99 %) for binary classification, and the DNN classifier had the best accuracy (99.57 %) for multiclass classification.

In (Awan et al., 2021), various machine learning models are used to predict real-time DDoS attacks at the application layer, utilizing Apache Spark, a distributed system, and a classification method to improve algorithm execution. The big data method's outcomes are compared to the non-big data approach, using the Scikit ML library and Spark-ML library with Random Forest (RF) and Multi-Layer Perceptron (MLP) machine learning techniques on the application layer DDoS dataset from Kaggle to detect DoS attacks. The MLP classifier using the non-big data approach has a minimum accuracy of 99.05%. In contrast, the RF classifier using the big data approach achieved a maximum accuracy of 99.94% and an F1-score of 99.95%. The proposed model's minimal processing time using an MLP classifier and big-data approach was 0.04 seconds. Limitations include only using two machine learning models and the dataset

having only two classes.

3 METHODOLOGY

This chapter describes the tools and methods utilized to develop IntrusionHunter. Fig 1 gives an overview of the entire process of detecting attacks accurately and efficiently. The entire process consists of three main phases: data cleaning (1), feature selection (2), and classification (3).

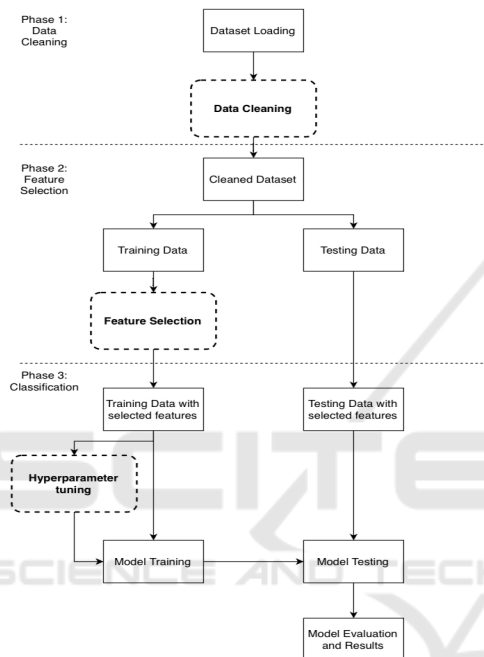


Figure 1: The three phases for intrusion detection by IntrusionHunter.

3.1 Dataset

The CSE-CIC-IDS2018 dataset (Communications Security Establishment (CSE), 2018) is a popular resource for creating and assessing intrusion detection systems. It contains over 10 million network traffic examples gathered over nine months in 2018, including a wide variety of malicious and benign traffic, with 14 attack classes categorized into Brute-force, Botnet, DoS, DDoS, infiltration, and Web attacks. The dataset's realism and diversity are among its essential characteristics, with traffic from various devices and networks. Each instance of network traffic is represented by a collection of attributes, including source and destination IP addresses, port numbers, protocol, and other specifics, which can be used to evaluate and train IDS systems' accuracy and false positive rate performance. The dataset also includes

labels describing each instance's traffic type, allowing for the evaluation of IDS systems' recall and precision.

3.2 Data Cleaning (Phase 1)

In phase 1, preprocessing steps are taken to ensure the dataset is clean and suitable for training. The raw dataset comprises ten separate CSV files, each containing network traffic records for a single day of operation. During dataset exploration, it was discovered that several column names were duplicated due to merging several CSV files. To resolve this issue, header-containing columns are deleted, and the data frame is exported to a temporary CSV file to be reread with appropriate column datatypes. The data is cleaned by dropping unnecessary columns, removing missing values and duplicates, and replacing "Infinity" with the mean value of the column. Fig 2 shows the process to ensure the data is clean and efficient for usage in the following steps.

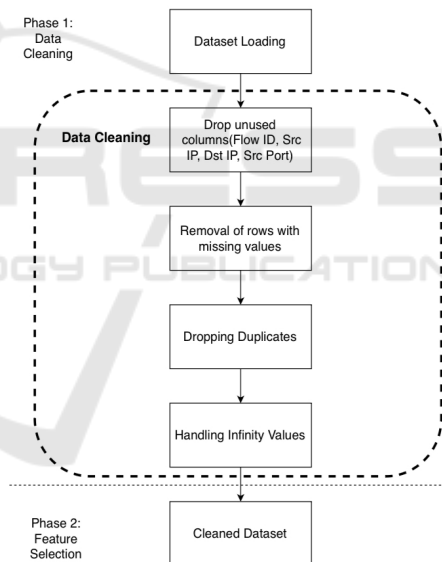


Figure 2: Data Cleaning Steps.

3.3 Feature Selection (Phase 2)

In phase 2, a subset of features is chosen from a dataset to employ in a machine learning model, enhancing performance and decreasing the risk of overfitting. Random forest is a popular technique for feature selection, initially trained on the dataset using several decision trees. The importance of each feature is assessed by calculating the decrease in impurity caused by splits using the feature, with each tree built using a random subset of the features. Averaging the scores across all the trees yields the final impor-

tance score for each feature. Features with a higher significance score are more beneficial for predicting the target variable, and 25 features were selected out of the 80 features in the dataset. Fig 3 explains the feature selection process, which is applied to the training data after splitting the dataset into training and testing data.

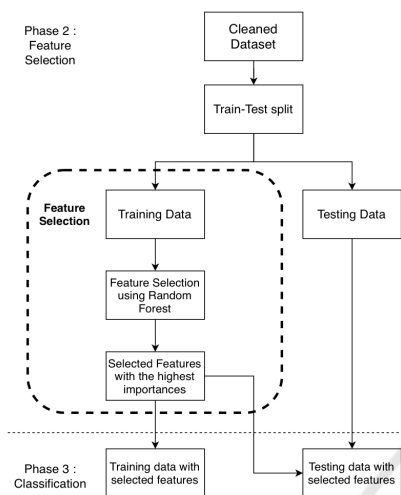


Figure 3: Feature Selection Phase.

3.4 Classification Models (Phase 3)

In phase 3, four primary classifiers are used for anomaly-based intrusion detection: Random forest, XGBoost, Convolutional Neural Networks, and Deep Neural Networks. Each model performs three types of classifications, as described in 3.4.1, and is referred to with a unique name consisting of the model’s abbreviation and the number of classes it classifies. For example, a random forest classifier that performs binary classification is referred to as (RF-2C). Fig 4 provides an overview of the code flow until testing the performance of the proposed intrusion detection system. After the data is split into training and testing data and feature selection is applied, the hyperparameter tuning process is performed to optimize the model’s hyperparameters and improve its performance. Optuna, a library that performs a search for the best hyperparameters by trying out different combinations and evaluating the model’s performance for each combination, is used to tune the model with 50 trials. Stratified sampling is used to take a sample of the dataset for hyperparameter tuning, as the dataset is excessively large and imbalanced. After hyperparameter tuning, the model is trained with the best-performing hyperparameters and the whole training data. Finally, the model is tested with the testing data, and the results are evaluated.

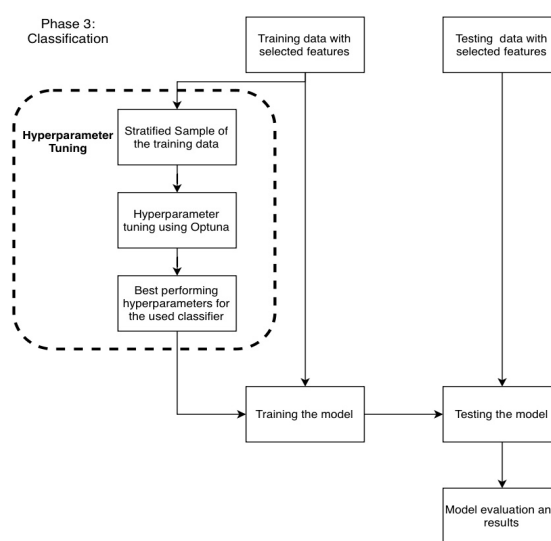


Figure 4: Model Classification.

3.4.1 Types of Classifications (2C-7C-15C)

After preprocessing and feature selection, various classification algorithms are applied to the data. The dataset undergoes binary classification (2C) and multiclass classification with seven (7C) and fifteen classes (15C). Binary classification is crucial in anomaly-based intrusion detection systems for identifying suspicious behavior indicative of security threats and classifying network traffic as benign or malicious. The multiclass classification is conducted in two ways: grouping similar attacks into seven main clusters (Benign, Brute-force, Botnet, DoS, DDoS, Infiltration, and Web attacks) to identify prevalent attack types, and classifying each of the original 15 attacks individually to enhance detection and prevention of specific attack types and recognize malicious patterns and trends.

3.4.2 Random Forest (2C, 7C, 15C)

Our study utilized the random forest algorithm as the first classifier. This algorithm is a popular choice for classification tasks due to its ability to handle large and imbalanced datasets, which is often the case in anomaly-based intrusion detection systems. Random forests construct multiple decision trees, each trained on a different subset of the data, to avoid bias towards the majority class. Additionally, the technique of bagging helps to avoid overfitting by training each tree on a random subset of the data. Random forests do not require data standardization before classification, making them a strong choice for our study.

Table 1 provides the parameters used in the random forest classifier. These parameters were returned

Table 1: Random Forest Parameters.

Parameter	RF-2C	RF-7C	RF-15C
number of estimators	150	170	200

after hyperparameter tuning was performed. It was noticed that when increasing the number of parameters to be tuned, the performance degraded. Therefore, `n_estimators` was the parameter chosen, as it greatly influences the classifier's performance.

3.4.3 Extreme Gradient Boosting (2C, 7C, 15C)

The XGBoost classifier, is a potent gradient boosting method, in addition to the random forest classifier. XGBoost builds a sequence of decision trees, each boosting the performance of the previous one. It has more capabilities than Random Forest, including better performance due to its optimization of the objective function, resulting in better predictions. XGBoost also includes built-in regularization techniques, such as L1 and L2 regularization and a dropout regularization technique, to prevent overfitting, a common problem with machine learning models.

Table 2: XGBoost Parameters.

Parameter	XGB-2C	XGB-7C	XGB-15C
number of estimators	1900	2000	2200
number of jobs	-1	-1	-1
tree method	'hist'	'hist'	'hist'

Table 2 includes the parameters of the XGBoost classifier. The number of estimators parameter was chosen using hyperparameter tuning, with the maximum value in each range selected. The `n_jobs` parameter specifies the number of CPUs used to train a model, with -1 using all available CPU cores to speed up training but potentially consuming more system resources. The `tree_method` parameter specifies the method for constructing decision trees, with 'hist' being an efficient histogram-based algorithm for high-dimensional data that can lead to faster training times than the traditional 'exact' method.

3.4.4 Convolutional Neural Networks (2C, 7C, 15C)

Convolutional neural networks (CNNs) are commonly used in computer vision applications, including object detection and image classification, and are well-suited for anomaly-based intrusion detection systems. Before training data using CNNs, standardization was performed to reduce the model's sensitivity to input feature scaling. Reshaping the 2D data into a 3D array allows us to use CNNs to extract features for classification. Dropout regulariza-

tion and batch normalization were used to prevent overfitting during training. Dropout randomly sets a fraction of input units to zero during each training epoch, while batch normalization normalizes the input of each layer to have zero mean and unit variance, stabilizing and accelerating the training process.

Table 3: CNN Layers.

Parameter	CNN-2C	CNN-7C	CNN-15C
Number of Conv1D layer	4	4	6
Number of MaxPooling layer	2	2	3
Number of Dropout layer	2	2	3
Number of BatchNormalization layer	2	2	3

As shown in table 3, a summary of the parameters used in three different convolutional neural networks (CNN) models are provided, labeled as "CNN-2C", "CNN-7C", and "CNN-15C". The table includes the number of Conv1D layers, MaxPooling layers, Dropout layers, BatchNormalization layers, activation function, optimizer, batch size, and number of epochs for each of the three models. The CNN-2C model uses 4 Conv1D layers, 2 MaxPooling layers, 2 Dropout layers, 2 BatchNormalization layers, 'relu' activation function, 'adam' optimizer, batch size 4096, and 50 epochs. The CNN-7C model uses 4 Conv1D layers, 2 MaxPooling layers, 2 Dropout layers, 2 BatchNormalization layers, 'relu' activation function, 'adam' optimizer, batch size 256 and 50 epochs. The CNN-15C model uses 6 Conv1D layers, 3 MaxPooling layers, 3 Dropout layers, 3 BatchNormalization layers, 'relu' activation function, 'adam' optimizer, batch size 4096, and 50 epochs.

3.4.5 Deep Neural Networks (2C, 7C, 15C)

As a final classification method in our IDS, deep neural networks (DNNs) are employed. DNNs are a form of artificial neural network that is made up of many layers of interconnected nodes. They can learn from new data by modifying the weights of the connections between the nodes. DNNs are widely employed in numerous applications and have been demonstrated to be exceptionally effective at classification jobs. As the CNNs, the data also need to be standardized before training. Both the performance and the behavior of the classifiers were enhanced by tuning their hyperparameters using a library named Optuna and some manual tuning. Unlike the CNN model, only drop out layers were used to prevent overfitting.

Table 4: DNN Layers.

Parameter	DNN-2C	DNN-7C	DNN-15C
Number of Dense layer	6	7	10
Number of Dropout layer	6	7	10

Table 4 displays the parameters of three deep neural network models (DNN-2C, DNN-7C, DNN-15C) used for classifying network intrusions. These models use dense layers as the primary building blocks, with the number of dense layers increasing from 2 to 7 and finally to 10 as we move from DNN-2C to DNN-15C. The activation function used in all three models is 'relu'. The models are optimized using the 'adam' optimizer and are trained using a batch size of 4096 samples for 50 epochs. The number of dropout layers in each model equals the number of dense layers. Dropout layers reduce overfitting by randomly setting some activations to zero during training.

4 EVALUATION AND RESULTS

This section evaluates the performance of the proposed anomaly-based intrusion detection system (IDS) models and compares them to state-of-the-art models using the same dataset. The dataset includes binary and multi-class classification tasks, and models are assessed using a variety of metrics 4.1, which is suitable for imbalanced datasets. Tables display each model's performance, with separate tables for each classification type. The aim is to showcase the effectiveness of the proposed models in anomaly detection and identify their strengths and weaknesses compared to state-of-the-art models.

4.1 Evaluation Metrics

The F1-score, a prevalent metric for evaluating anomaly-based intrusion detection systems, is the harmonic mean of precision and recall, where higher values signify better performance. Precision quantifies the ratio of accurate positive predictions, while recall assesses the model's capacity to identify all true positive instances. The F1-score is useful for IDS as it enables the evaluation of both false positives and false negatives, which have significant consequences (Cardenas et al., 2006). The formulas to calculate F1-score 1, precision 2, recall 3, and accuracy 4 are provided.

$$F1 - score = 2 * \frac{precision * recall}{precision + recall} \quad (1)$$

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

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

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

True positive (TP): The number of instances correctly classified as positive.

True negative (TN): The number of instances correctly classified as negative.

False positive (FP): The number of instances incorrectly classified as positive.

False negative (FN): The number of instances incorrectly classified as negative.

4.2 Binary Classification (2C)

Table 5: Binary Classification Results.

Study	Accuracy	F1-score	Approach
CNN-2C	97.997%	97.939%	CNN
RF-2C	99.332%	99.328%	Random Forest
XGB-2C	99.036%	99.041%	XGBoost
DNN-2C	94.667%	94.234%	DNN
RF-2C-Botnet	99.999%	99.996%	Random Forest
XGB-2C-Botnet	99.911%	99.996%	XGBoost
Kanimozhi and Prem Jacob (Kanimozhi and Jacob, 2019)	99.97%	99.91%	ANN
Praneeth (Praneeth et al., 2021)	99.57%	98%	DNN
Seth, Singh, and Chahal (Seth et al., 2021)	97.73%	97.73%	LightGBM

Table 5 displays the binary classification outcomes of various anomaly-based intrusion detection systems (IDS), including CNN-2C, RF-2C, XGB-2C, DNN-2C, RF-2C-Botnet, XGB-2C-Botnet, Kanimozhi and Prem Jacob (Kanimozhi and Jacob, 2019), Praneeth (Praneeth et al., 2021), Seth, Singh, and Chahal (Seth et al., 2021). RF-2C achieves the highest accuracy and F1-score for classifying the entire dataset, with 99.332% and 99.328%, respectively. XGB-2C has the second-highest accuracy and F1-score, with 99.036% and 99.041%, respectively. RF-2C-Botnet and XGB-2C-Botnet outperform Kanimozhi and Prem Jacob's model (Kanimozhi and Jacob, 2019) for classifying botnet attacks, with accuracies of 99.999% and 99.911% and F1-scores of 99.996% and 99.996%, respectively. DNN-2C has the lowest accuracy and F1-score, while Seth, Singh, and Chahal's model (Seth et al., 2021) has the second-lowest. It is worth noting that Praneeth's IDS (Praneeth et al., 2021) only uses a subset of the dataset.

4.3 Multiclass Classification (7C)

Table 6: Multiclass Classification (7 Classes) Results.

Study	Accuracy	F1-score	Approach
CNN-7C	97.814%	97.206%	CNN
RF-7C	99.240%	99.135%	Random Forest
XGB-7C	98.949%	98.626%	XGBoost
DNN-7C	94.244%	93.215%	DNN
Lin, Ye, and Xu (Lin et al., 2019)	96.2%	85%	LSTM
Karatas, Demir, and Sahingoz (Karatas et al., 2020)	95.49%	99.7%	Adaboost

Table 6 presents the results of six different anomaly-based intrusion detection systems (IDSs) applied to classify network intrusions into seven classes. The CNN-7C, RF-7C, XGB-7C, and DNN-7C IDSs use the CNN, Random Forest, XGBoost, and DNN approaches. The table also includes the results of

two additional IDSs proposed by Lin, Ye, and Xu (Lin et al., 2019) and Karatas, Demir, and Sahingoz (Karatas et al., 2020), which use the LSTM and Adaboost with gradient boosting approaches, respectively. The CNN-7C IDS has an accuracy of 97.814%, while the RF-7C IDS has the highest accuracy at 99.240%. The F1-score is also the highest for the RF-7C IDS at 99.135%. The XGB-7C and DNN-7C IDSs have lower F1-score compared to their accuracy values. The LSTM-based IDS proposed by Lin, Ye, and Xu (Lin et al., 2019) has a relatively high accuracy of 96.2%, but a lower F1-score of 85%. The Adaboost and gradient boosting IDS proposed by Karatas, Demir, and Sahingoz (Karatas et al., 2020) has a high F1-score of 99.7%, but a lower accuracy of 95.49%. Overall, among the seven-class IDSs, the RF-7C IDS appears to have the best balance of accuracy and F1-score.

4.4 Multiclass Classification (15C)

Table 7: Multiclass Classification (15 Classes) Results.

Study	Accuracy	F1-score	Approach
CNN-15C	97.313%	96.643%	CNN
RF-15C	99.239%	99.124%	Random Forest
XGB-15C	98.951%	98.628%	XGBoost
DNN-15C	95.152%	94.077%	DNN
Ferrag (Ferrag et al., 2020)	97.376%	-	CNN

Table 7 displays the outcomes of several anomaly-based intrusion detection systems for multiclass classification with 15 classes, including CNN, Random Forest, XGBoost, and DNN. RF-15C has the highest accuracy with 99.239% and the highest F1-score with 99.124%. DNN-15C has the lowest accuracy with 95.152% and the lowest F1-score with 94.077%. The other models have accuracy and F1-score scores between these two extremes. Ferrag's study (Ferrag et al., 2020) uses a CNN model, achieves an accuracy of 97.376%, and does not state the F1-score. Overall, the CNN, Random Forest, and XGBoost models perform well in terms of accuracy, but RF-15C and XGB-15C have the best F1-score. The DNN model has a moderate performance, with relatively lower accuracy and F1-score than the other models.

4.5 Discussion

The performance of anomaly-based intrusion detection systems (IDSs) varies across classification tasks and approaches. In binary classification (Table 5), Random Forest (RF-2C) and XGBoost (XGB-2C) IDSs outperform other models like DNN (DNN-2C) with the highest accuracy and F1-score values (99.332% and 99.328%, respectively). For seven-

class classification (Table 6), the Random Forest (RF-7C) IDS achieves the highest accuracy and F1-score values (99.240% and 99.135%, respectively). In 15-class classification (Table 7), the Random Forest model (RF-15C) achieves the highest accuracy and F1-score (99.239% and 99.124%, respectively). Tree-based models, particularly Random Forest and XGBoost, consistently perform well across binary and multiclass classification tasks in intrusion detection systems. Deep learning models, such as CNNs and DNNs, show lower performance compared to tree-based models, which may be due to dataset imbalance. Neural networks may learn to predict the majority class more accurately at the expense of the minority class as they are optimized to minimize the overall error, which is dominated by the majority class (Shin et al., 2016). Additionally, the F1-score is a more reliable metric than accuracy when evaluating the performance of an intrusion detection system because it takes into account both precision and recall. Accuracy, on the other hand, only measures the proportion of correct classifications out of all instances, which can be misleading in the context of intrusion detection.

5 CONCLUSION AND FUTURE WORK

This section will analyze the study's primary findings, limitations of the classification models used, and future work on anomaly-based intrusion detection.

5.1 Conclusion

This paper evaluates the performance of the presented IDS (IntrusionHunter) using the CSE-CICIDS2018 dataset (Communications Security Establishment (CSE), 2018). Data cleaning was performed to ensure high-quality data, including removing missing or invalid values. Feature selection was done using the random forest approach, and four classification approaches were used: random forest, XGBoost, deep neural network, and convolutional neural network. The IDSs were evaluated using binary, multiclass classification with 7 classes, and multiclass classification with 15 classes. The Random Forest IDSs outperformed the other approaches, with superior accuracy and F1-score values in binary classification (2C) and multiclass classification (15C) tasks, and favorable results in the multiclass classification (7C) task. The CNN and DNN approaches did not perform as well, mainly due to dataset imbalance, resulting in lower accuracy and F1-score values. Fur-

ther research is needed to improve the performance of IDSs and develop approaches that can effectively classify network intrusions into different classes.

5.2 Future Work

The study's limitations include the small dataset size that may not represent all attacks and an imbalanced dataset. The classification models were only evaluated on one dataset, and testing them on additional datasets would be beneficial. Future work aims to address the imbalance issue, evaluate the models on more datasets, and compare their performance across various data types. The study also plans to explore other classification models, such as recurrent neural networks and long short-term memory networks. Additionally, integrating the results of multiple classification models into a single anomaly detection system could potentially enhance its overall performance.

REFERENCES

- Alqahtani, H., Sarker, I., Kalim, A., Minhaz Hossain, S., Ikhlaq, S., and Hossain, S. (2020). Cyber intrusion detection using machine learning classification techniques. In Chaubey, N., Parikh, S., and Amin, K., editors, *Computing Science, Communication and Security*, Communications in Computer and Information Science, pages 121–131, United States. Springer, Springer Nature. 1st International Conference on Computing Science, Communication and Security, COMS2 2020 ; Conference date: 26-03-2020 Through 27-03-2020.
- Awan, M., Farooq, U., Babar, H., Yasin, A., Nobanee, H., Hussain, M., Hakeem, O., and Zain, A. (2021). Real-time ddos attack detection system using big data approach. *Sustainability*, 13:10743.
- Burbank, J. L. (2008). Security in cognitive radio networks: The required evolution in approaches to wireless network security. In *2008 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, pages 1–7.
- Cardenas, A., Baras, J., and Seamon, K. (2006). A framework for the evaluation of intrusion detection systems. In *2006 IEEE Symposium on Security and Privacy (S&P'06)*, pages 15 pp.–77.
- Communications Security Establishment (CSE), C. I. f. C. C. (2018). Cse-cicids2018 dataset. <https://www.unb.ca/cic/datasets/ids-2018.html>.
- Dowd, P. and McHenry, J. (1998). Network security: it's time to take it seriously. *Computer*, 31(9):24–28.
- Faker, O. and Dogdu, E. (2019). Intrusion detection using big data and deep learning techniques. In *Proceedings of the 2019 ACM Southeast Conference*, ACM SE '19, page 86–93, New York, NY, USA. Association for Computing Machinery.
- Ferrag, M. A., Maglaras, L., Moschoyiannis, S., and Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50:102419.
- for Cybersecurity (CIC), C. I. (2018). Intrusion detection evaluation dataset (cic-ids2017). <https://www.unb.ca/cic/datasets/ids-2017.html>.
- He, J., Yang, J., Ren, K., Zhang, W., and Li, G. (2019). Network security threat detection under big data by using machine learning. *Int. J. Netw. Secur.*, 21:768–773.
- Information and of California, C. S. U. (1999). Kdd cup 99 dataset. <http://kdd.ics.uci.edu/databases/kddcup99>.
- Kanimozhi, V. and Jacob, T. P. (2019). Artificial intelligence based network intrusion detection with hyperparameter optimization tuning on the realistic cyber dataset cse-cic-ids2018 using cloud computing. In *2019 international conference on communication and signal processing (ICCSP)*, pages 0033–0036. IEEE.
- Karatas, G., Demir, O., and Sahingoz, O. K. (2020). Increasing the performance of machine learning-based idss on an imbalanced and up-to-date dataset. *IEEE Access*, 8:32150–32162.
- Lin, P., Ye, K., and Xu, C.-Z. (2019). Dynamic network anomaly detection system by using deep learning techniques. In *International conference on cloud computing*, pages 161–176. Springer.
- Marchang, N., Datta, R., and Das, S. K. (2017). A novel approach for efficient usage of intrusion detection system in mobile ad hoc networks. *IEEE Transactions on Vehicular Technology*, 66(2):1684–1695.
- Praneeth, V., Kumar, K. R., and Karyemsetty, N. (2021). Security: intrusion prevention system using deep learning on the internet of vehicles. *International Journal of Safety and Security Engineering*, 11(3):231–237.
- Seth, S., Singh, G., and Kaur Chahal, K. (2021). A novel time efficient learning-based approach for smart intrusion detection system. *Journal of Big Data*, 8(1):1–28.
- Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., and Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35(5):1285–1298.
- Sydney, U. Unsw-nb15 dataset. <https://research.unsw.edu.au/projects/unsw-nb15-dataset>.