# An Intrusion Detection System for Multi-Class Classification based on Deep Neural Networks

Petros Toupas, Dimitra Chamou, Konstantinos M. Giannoutakis, Anastasios Drosou, Dimitrios Tzovaras
*Information Technologies Institute*
*Center for Research & Technology-Hellas*
6th km Charilaou-Thermi, 57001, Thessaloniki, Greece
{ptoupas,dimicham,kgiannou,drosou,Dimitrios.Tzovaras}@iti.gr

***Abstract*—Intrusion Detection Systems (IDSs) are considered as one of the fundamental elements in the network security of an organisation since they form the first line of defence against cyber threats, and they are responsible to detect effectively a potential intrusion in the network. Many IDS implementations use flow-based network traffic analysis to detect potential threats. Network security research is an ever-evolving field and IDSs in particular have been the focus of recent years with many innovative methods proposed and developed. In this paper, we propose a deep learning model, more specifically a neural network consisting of multiple stacked Fully-Connected layers, in order to implement a flow-based anomaly detection IDS for multi-class classification. We used the updated CICIDS2017 dataset for training and evaluation purposes. The experimental outcome using MLP for intrusion detection system, showed that the proposed model can achieve promising results on multi-class classification with respect to accuracy, recall (detection rate), and false positive rate (false alarm rate) on this specific dataset.**

***Index Terms*—Cybersecurity, Intrusion Detection System, Deep Neural Networks, CICIDS2017, Flow Feature-Based, Multi-Class Classification**

## I. Introduction

During the past few years, the rising exposure of many organisations to sophisticated cyber-attacks have led to a rapid development of innovative IDSs. The development of IDSs concerns both the academic and the industrial community worldwide, due to the impact that each cyber attack has, as economic cost, reputational damage, and legal sequences. Therefore, it is a matter of great importance to secure networks from unauthorized access and protect the user communication and their data, [1], as well as to reveal new security issues that arise.

### A. Intrusion Detection System

Intrusion Detection System (IDS) is an efficient security reinforcement tool for the detection and the protection of cyber-attacks in any network or host. The IDSs‘ responsibility is to detect suspicious behaviors and act appropriately to protect the network from the onset of attacks and reduce functionally and financial losses, [2].

In literature, IDSs can be categorized as, [3], either signature-based, [4], anomaly-based, [5], or a hybrid combination of both.

Signature-based intrusion detection systems (SIDS), also known as Rule-based or Misuse IDS, conduct ongoing monitoring of network traffic and seek out sequences or patterns of inbound network traffic that matches an attack signature. They work with high accuracy rates in identifying possible known invasions, by keeping error rates low. One of the drawbacks of these systems is that there has to be an up to date database containing the attacks signatures.

The anomaly-based intrusion detection systems (AIDS), or behavior-based detection, analyzes the normal network's behavior, by monitoring network traffic to detect abnormal activity. AIDS have the ability to be trained with anomaly detection algorithms or to be self-trained with self-learning algorithms, so they can detect new types of intrusions. Compared to signature-based, anomaly-based shows a significant difference in identifying novel attacks.

Hybrid Intrusion Detection System (HIDS) can combine the advantages of both signature-based and anomaly-based system and increase the detection of known intrusion attacks, while eliminating the error rates of unknown attacks. Most of the latest hybrid IDSs are based on machine and deep learning methods.

### B. Flow feature-based Classification

One of the main methods for intrusion detection is the network traffic analysis and the extraction of various statistical features in order to detect abnormal network traffic, in near-real time. IDSs, in order to work properly and detect abnormal activities effectively, use network flows created based on source/ destination IP, source/ destination port, protocol, and timestamp, [6], [7]. A useful flow definition is mentioned bellow. *A flow is a group of IP packets with some common properties passing a monitoring point in a specified time interval*, [8].

Cisco, [9], referred that *A complete flow is a unidirectional exchange of consecutive packets on the network between a port at an IP address and another port at a different IP address, using a particular application protocol.*

Therefore, traffic classification is necessary for the efficient flow management, processing and machine learning exploitation, [10]. In general, the most widespread and broad traffic classification categories are using different flow features and

978-1-7281-4550-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMLA.2019.00206

are divided to port-based methods, payload-based methods, host-based and flow feature-based methods.

Intrusion detection systems apply different anomaly detection methods, depending on each case study, the available resources and the accessible technologies. The current work focuses on the flow feature-based technique, since it can overcome numerous limitations of other techniques, such as unregistered port numbers, encrypted packet payload etc. Flow-based method uses flow features as discriminators to exploit the diversity of the traffic packets and map flows to classes, [10]. Moreover, concerning the privacy issues, flow-based method is preferable instead of payload method, because of the absence of payload. A comprehensive flow-based intrusion detection survey conducted by Sperotto et al., can be found here [11].

### C. Machine Learning and Deep Learning techniques for IDS

Machine learning and deep learning techniques have been used to develop IDSs in the field of cybersecurity. In order to increase effectiveness of IDSs, the research has been focused on novel learning technologies and algorithms of Artificial Neural Networks (ANN), Support Vector Machines (SVM), Naive-Bayesian (NB), Random Forests (RF), self-organizing map (SOM) etc.

The need for a complete, rich, up-to-date, and well-formed dataset with various criteria and features, is a key concern of researchers for experiments conduction, testing, and evaluation of the models, [12], [13] on modern networks. A dataset is appropriate when it is *updated in time due to the high malware mutation and evolution*, *represents real world network traffic*, *has traffic diversity and volume*, and *is publicly available* In literature, there are numerous datasets available for experimentation, but only a few fulfil all the desired features. It is advisable to refer to some of the most well-known, nominally, with few details.

KDD-99, [14], CAIDA, [15], ISCX2012, [16], Kyoto, [17] are datasets that represent real world network traffic, but nowadays, are considered outdated due to the continuous evolution of network‘s attacks and threats. On the other hand, a quite popular in the field of research is the CICIDS2017 dataset, which was released recently and contains many traffic types, fulfils the criteria of the real world network traffic, and was created to overcome some issues of existing datasets. Based on Gharib et al., [18], CICIDS2017 meets all the criteria of an accurate and complete dataset. Based on this dataset we have implemented a deep neural network model to classify the network traffic based on statistics produced by network flow analysis.

The rest of the paper is organized as follows. Section II presents related work on network intrusion detection using machine learning and deep learning techniques. Section III summarizes the data analysis and the pre-processing procedure. In Section IV, we provide a description of the proposed architecture of the Deep neural network. In Section V, we present the results of our work comparing it with related works. Finally, Section VI gives a conclusion of this paper and presents future work.

## II. Related Work

In recent literature, most of the studies in flow-based Intrusion Detection Systems based on machine-learning technologies are using the CICIDS2017 dataset, for the training and the evaluation. However, due to the new entrant dataset in the field of cyber security, there are limited published studies yet.

Ullah and Mahmou, [19], proposed a hybrid model anomaly detection model, based on flow-based anomaly detection for the classification at CICIDS2017 and UNSW-15 datasets. They used Recursive Feature Elimination (RFE) for the selection of significant features, Synthetic Minority Over-Sampling Technique (SMOTE) for the oversampling and Edited Nearest Neighbors (ENN) for the cleaning of the dataset. At level-1 the network flows were classified with decision tree classifier, as normal or abnormal, and were forwarded to the level-2 in order to determine the type of the attack (multi-classification). The results of specificity, precision, recall and F-score for level-1 were measured 100% for the CICIDS2017 dataset and 99% for the UNSW-15 dataset, while the level-2 model precision, recall, and F-score were measured at 100 % for the CICIDS2017 dataset and 97 % for the UNSW-15 dataset, respectively.

Vijayanand, Devaraj and Kannapiran, [20], proposed a novel IDS with genetic-algorithm-based feature selection and multiple support vector machine classifiers for wireless mesh networks. In order to succeed better accuracy, they select specific features exploiting the Genetic Algorithm-based feature selection and SVM classifier. The evaluation of the system is done using an intrusion dataset, generated from a WMN, and simulated in Network Simulator 3 (NS3) tool by using the standard intrusion dataset. Moreover, they validate the system using ADFA-LD and CICIDS2017 intrusion datasets. A comparative analysis is performed, between the proposed system and MI-based feature selection, suggesting that GA-based feature selection with SVN classifier exhibit better performance metrics, with higher accuracy, about 99%, and less computational complexity.

Zhang, et al., [2], presented a anomaly detection model based on neural network. They designed an IDS using LeNet-5 convolutional neural network and LSTM network for feature extraction. The experiments were conducted using the CICIDS2017 and CTU datasets for both binary and multi-classification. They performed CNN, LSTM and the hybrid combination of both, which achieved good classification results in both binary-classification and multi-classification experiments. The accuracy succeeded was about 99%. They, also analyzed the flows which were important for the classification and for the efficient abnormal detection.

Ferrag and Maglaras, [21], presented the DeepCoin which is a novel deep learning and blockchain-based energy framework to protect the smart grid against cyber attacks. The used the practical Byzantine fault tolerance algorithm recurrent neural network algorithm for the block-based network using deep learning. They worked in three different datasets for evaluation reasons and performance testing, including CICIDS2017, a power system dataset, and a web robot (Bot)-Internet of Things dataset. The accuracy rate by using recurrent neural networks, with backpropagation through time was 98.23%.

Binbusayyis and Vaiyapuri, [22], mainly focused on creating an ensemble for feature selection using different evaluation measures, that can implement an intrusion detection system. Particularly, they proposed a set of feature selection and feature extraction and developed an IDS model by using the learning algorithm, Random Forest. The evaluation was done on various evaluation datasets, namely, KDDCup'99, NSL-KDD, UNSW-NB15 and CICIDS2017, in order to demonstrate the effectiveness of the proposed model. The results revealed that the specific subset of features is promising due to the final high performance metrics, achieving 99.88% accuracy, compared to other approaches.

Radford, Richardson and Davis, [23], presented an anomaly detection sequence model based on Long Short-term Memory (LSTM) recurrent neural network (RNN). They used embedded sequences passed through two bidirectional LSTM models in order to implement the proposed system. The testing experiments were conducted with the use of CICIDS2017 and the model results aimed at multi-classification.

## III. Pre-Processing and Data Analysis

### A. CICIDS2017 Dataset Description

This work is relying on a public intrusion detection dataset namely CICIDS2017, [13], created by the University of New Brunswick (UNB) in cooperation with the Canadian Institute for Cybersecurity (CIC). The CICIDS2017 dataset not only contains the most up to date network attack scenarios but also fulfills all the criteria of real-world cyber attacks.

The dataset contains benign (normal) and abnormal (different types of attacks) network traffic from five consecutive days of capturing, and it is divided into 8 different files. For each day a different type of attack was deployed as described in [13]. The class distribution of the dataset is also presented in [13].

Each row in the dataset contains 83 features which have been extracted from the network traffic using the CICFlowMeter tool, [24], [25]. The CICFlowMeter generates Bidirectional Flows (Biflow), where the first packet determines the forward (source to destination) and backward (destination to source) directions, hence the 83 statistical features include data that derive from both the forward and reverse direction. We used a subset of the original 83 features, ommiting some features like source and destination IPs, the ID of the Biflow, timestamp etc., and we ended up with a dataset of 79 features.

### B. Data Cleansing

Machine learning algorithms are directly related to data, and in order to be as accurate as possible, this data needs to be refined. Firstly we identified rows/Biflows in the dataset having missing values, infinity values, and values that did not make sense (i.e. negative time duration of a communication, etc.). This step is crucial in order to maintain the reliability of the dataset and not to add noise, so the choice of method has to be done with caution. In our case, most of the rows with missing/wrong values were on classes with many examples (BENIGN, DoS Hulk, PortScan, DDoS) so we decided to remove them, since we already had enough examples to work with.

Moreover, we discarded the features with zero variance (i.e. features with a constant value for all the examples), since they could not provide additional statistical information, [26], for our ML algorithm to be able to "learn" from these features.

We also conducted a Pearson, [27]–[29] correlation test on the remaining features in order to evaluate the associations between them. If two or more features are highly correlated, this implies that they are measuring the same underlying information, so removing one should not compromise the performance of the model and may even lead to better results. Pearson correlation coefficient or Pearson's **r** is the metric which measures the linear correlation between two variables and it's value lays in $[-1, 1]$. We have removed features where this metric was above or below the threshold of 0.95 or -0.95 respectively.

Finally, we checked and deleted all the duplicate rows/Biflows. As a result of the above cleaning and feature extraction methods we end up with a dataset of 2515416 examples and 45 features.

### C. Data Transformation

We have decided to merge three of the dataset classes into one larger common class. These classes are Web Attack-Brute Force, Web Attack-XSS, and Web attack Sql Injection. We merged them as their behavior in network traffic level is almost identical (something that was also confirmed by the results of several different ML models during the evaluation phase). The final distribution of the classes on the cleansed dataset is show in Table I.

TABLE I
Class distribution of CICIDS2017 Dataset

| ID | Label | Number of Instances | % w.r.t. the number of total instances |
|---|---|---|---|
| 1 | BENIGN | 2089692 | 83.07 |
| 2 | DoS Hulk | 172838 | 6.87 |
| 3 | PortScan | 128008 | 5.08 |
| 4 | DDoS | 90694 | 3.6 |
| 5 | Dos GoldenEye | 10283 | 0.41 |
| 6 | FTP-Patator | 5931 | 0.23 |
| 7 | SSH-Patator | 5385 | 0.21 |
| 8 | DoS slowloris | 5228 | 0.20 |
| 9 | DoS Slowhttptest | 3219 | 0.12 |
| 10 | Web Attacks | 2143 | 0.085 |
| 11 | Bot | 1948 | 0.08 |
| 12 | Infiltration | 36 | 0.0014 |
| 13 | Heartbleed | 11 | 0.0004 |

Before start feeding the cleansed data into our ML algorithm we did some more statistical analysis by plotting and visually inspecting the distribution of each feature. During this process we noticed that many of the features were highly skewed (mostly on the left). There are many ways to deal with data skewness such as cube root, square root, logarithm transformation, or square, cube, or a higher power transformation. Although these methods work on many cases, we used the Yeo-Johnson, [30] transformation because it worked better for our ML algorithm.

Yeo-Johnson is a family of transformations which is appropriate for reducing skewness and to approximate normality. It extends the functionality of the original Box & Cox, [31] transformations, which is valid only for positive values, to be able to use both negative and positive values. The Yeo-Johnson transformations are defined as follows:

$$\psi(\lambda, x) = \begin{cases} ((y+1)^{\lambda} - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y+1)^{2-\lambda} - 1)]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 0, y < 0 \end{cases} \tag{1}$$

In Fig. 1 we present the effect of the Yeo-Johnson transformation on the distributions of some of the features in the CICIDS2017 dataset. The Yeo-Johnson transformation also take cares of the normalization of the data.

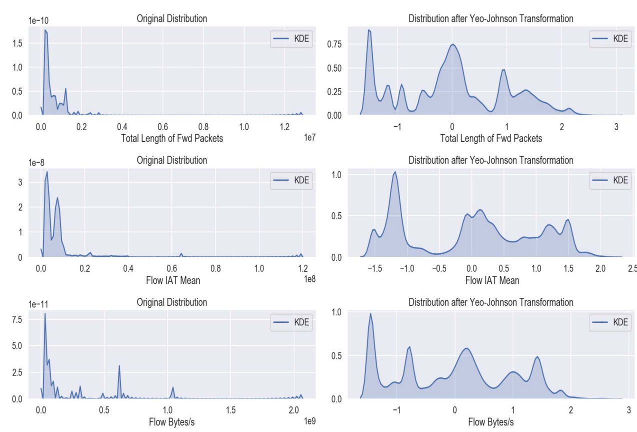



Fig. 1. Yeo-Johnson Transformation

The high class imbalance in the dataset is easy noticable from Table I. It can be very difficult for a ML algorithm to be able to "learn" how to map the input features of classes with very few examples to their corresponding labels since there are not many data to learn from. One of the most direct ways for dealing with class imbalance is to alter the class distributions toward a more balanced distribution. There are two basic methods for balancing class distributions, under-sampling and over-sampling.

There are many ways to under-sample and over-sample an imbalanced dataset, where some of the most common are described by Gustavo Batista et al. in, [32] and, [33]. In our case we have used the SMOTEENN method, which is a combination of the well known Synthetic Minority Over-Sampling Technique (SMOTE), [34] method for over-sampling the minority class (in our case there were more than one minority classes), followed by the Edited Nearest Neighbor (EEN), [35] method for under-sampling not only the majority class but all of the classes as a data cleaning method.

When using over-sampling methods, someone must be very careful with the evaluation process. If the over-sampling is performed before the splitting of the dataset in training, development, and test sets, then the evaluation will not be reliable since the test set has been merged with new artificially created data and its distribution will be different from the original.

## IV. Deep Neural Network Architecture

In the proposed architecture for the deep neural network, the model consists of one input layer which has 44 features passed as input to the neural network as those emerged from the feature engineering described in chapter III. The input layer is followed by 8 hidden layers with 140, 120, 100, 80, 60, 40, 20, and 120 nodes respectively. The final layer is the output layer or softmax layer, which produces the probabilities for the 13 classes where the prediction takes place.

For the initialization of the weights on Fully Connected layers we used the lecun-uniform initialization, [36], while for the output / softmax layer we used the glorot-uniform initialization, [37]. We came up using the ReLU, [38] as the activation function for all of the Dense layers. In order to train the neural network we used the Adam optimizer, [39]. Although regularization techniques like L1, L2 regularization, dropout are used quite often to address overfitting in neural networks, in our case it made no different to the final results so we did not use any regularization.

## V. Numerical Results

In this chapter we present the results of the proposed architecture in terms of recall, precision, and F1 score for each one of the 13 different classes that our model is able to detect. The evaluation of the model was based on a 10 fold cross validation. The metrics we used to evaluate our model in each one of the 10 splits of cross validation are all based on the confusion matrix that each of the splits produced. The metrics are accuracy, precision, recall, F1 score and false positive rate.

For each one of the metrics we have calculated the (macro) average over the 10 splits of the cross validation in order to be as much robust and precise as possible in the evaluation of our model.

The results (average of 10 splits) of Precision, Recall, F1 are presented in Fig. 2. Based on the results in Fig. 2 we have also calculated the averages over all of the classes of the model. The overall accuracy of our model is 99.95%, precision equals to 94.31%, recall or detection rate is 95.62%, and F1 Score is 94.1%. Beside that we have also calculated the False Positive Rate or False Alarm Rate, which is equal to 0.0005, as an average of the FPR of all classes and all the splits.

Finally, we have calculated the ROC Curves for each class, as long as the micro and macro averages of these curves. This can be shown in Fig. 3 alongside with the Area Under Curve (AUC) metric for each class and the average for all of the classes. The (macro) average AUC value of all the classes is equal to 0.99.

Comparison with relevant literature based on CICIDS2017 dataset, can not be performed directly, especially in the case of multi-class classification. Although, in most of the cases the evaluation details are not defined explicitly, a qualitative comparison can be performed. For example, Vijayanand et al., [20], used an SVM classifier resulting in a multi-class classification accuracy equal to 99.85% and FPR equal to 0.0009, but it is not stated if this was a result of a cross validation evaluation or if it was a random split of the dataset. The same occurs in some more cases, like in, [21], where Ferrag et. al. using an RNN classifier came up with the following results: accuracy of 99.81%, FPR of 0.009, and detection rate of 94.09%. Similarly, in, [22] and, [23] the authors using random forest and LSTM respectively came up evaluating their models using the AUC metric reporting 0.96 and 0.87 values as an average of all classes used. In, [2], Zhang et al., using CNN and LSTM models report accuracy equal to 99.77%, precision equal to 99.94%, recall equal to 99.95%, and F1 Score equal to 99.94%, classifying only 10 classes (dropping the ones with the fewer instances in the dataset) and without reporting the use of cross validation or not.

Even though the comparison with related work is not a straight forward process, the proposed work performs efficiently even though the relatively small model used for the classification.

## VI. Conclusions and Future Work

In this work we implemented a deep neural network model which is able to detect abnormal or malicious behavior (a potential cyber-attack) in the network traffic, and to classify the type of traffic between 13 different cases. During the data analysis and pre-processing of the dataset we were able to significantly reduce the number of input features to the model without reducing its performance at all. The proposed model achieves very promising results in a multi-class classification problem, while being at the same time a very simple and relatively small model.

As future work we plan to perform an analysis on reducing even further the input features, by testing various techniques such as Autoencoders or Principal Component Analysis (PCA), Independent Component Analysis (ICA), etc. so as not to reduce the performance of the model. One more thought is to improve and extend the current dataset with even more types of network based attacks and re-train the model to be able to detect them without reducing its performance on the original 13 classes. Finally, we plan on trying different types of architectures to approach this problem. More precisely, we consider the implementation of an RNN (LSTM, GRU, etc.) or CNN architectures for the model since the dataset contains sequential data.


## Acknowledgment

This work has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation, with Title H2020-FORTIKA "cyber-security Accelerator for trusted SMEs IT Ecosystem" under grant agreement no 740690.


## References


[1] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on sdn based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, Mar 2019. [Online]. Available: https://doi.org/10.1007/s12083-017-0630-0

[2] Y. Zhang, X. Chen, L. Jin, X. Wang, and D. Guo, "Network intrusion detection: Based on deep hierarchical network and original flow data," *IEEE Access*, vol. 7, pp. 37 004–37 016, 2019.

[3] Q. Zhou and D. Pezaros, "Evaluation of machine learning classifiers for zero-day intrusion detection - an analysis on CIC-AWS-2018 dataset," *CoRR*, vol. abs/1905.03685, 2019. [Online]. Available: http://arxiv.org/abs/1905.03685

[4] H. Holm, "Signature based intrusion detection for zero-day attacks: (not) a closed chapter?" in *2014 47th Hawaii International Conference on System Sciences*, Jan 2014, pp. 4895–4904.

[5] V. Jyothsna, V. V. Rama Prasad, and K. Munivara Prasad, "A review of anomaly based intrusion detection systems," *International Journal of Computer Applications*, vol. 28, pp. 26–35, 08 2011.

[6] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang, "Internet traffic classification by aggregating correlated naive bayes predictions," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 5–15, Jan 2013.

[7] Y. Zhang, X. Chen, L. Jin, X. Wang, and D. Guo, "Network intrusion detection: Based on deep hierarchical network and original flow data," *IEEE Access*, vol. 7, pp. 37 004–37 016, 2019.

[8] J. Quittek, T. Zseby, B. Claise, and S. Zander, "Requirements for ip flow information export (ipfix)," *RFC*, vol. 3917, pp. 1–33, 2004.

[9] C. Systems, "Cisco ios netflow," 2012.

[10] R. Boutaba, M. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. Caicedo Rendon, "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, 05 2018.

[11] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of ip flow-based intrusion detection," *IEEE Communications Surveys and Tutorials*, vol. 12, pp. 343–356, 09 2010.

[12] R. Panigrahi and S. Borah, "A detailed analysis of cicids2017 dataset for designing intrusion detection systems," vol. 7, pp. 479–482, 01 2018.

[13] I. Sharafaldin, A. Habibi Lashkari, and A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," 01 2018, pp. 108–116.

[14] C. Thomas, V. Sharma, and N. Balakrishnan, "Usefulness of darpa dataset for intrusion detection system evaluation," 03 2008.

[15] J. Udhayan and H. Thiag, "Statistical segregation method to minimize the false detections during ddos attacks," vol. 13, 11 2011.

[16] A. Shiravi, H. Shiravi, M. Tavallaee, and A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers Security*, vol. 31, p. 357–374, 05 2012.

[17] D. Protic, "Review of kdd cup '99, nsl-kdd and kyoto 2006+ datasets," *Vojnotehnicki glasnik*, vol. 66, pp. 580–596, 07 2018.

[18] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An evaluation framework for intrusion detection dataset," in *2016 International Conference on Information Science and Security (ICISS)*, Dec 2016, pp. 1–6.

[19] I. Ullah and Q. H. Mahmoud, "A two-level hybrid model for anomalous activity detection in iot networks," in *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2019, pp. 1–6.

[20] V. Anand, D. Devaraj, and B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Computers Security*, vol. 77, 04 2018.

Fig. 2. Model Evaluation on 10 Fold Cross Validation

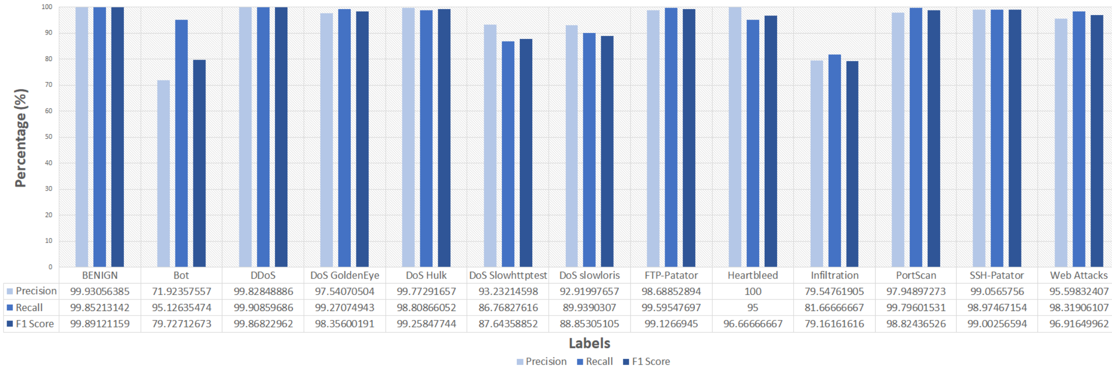
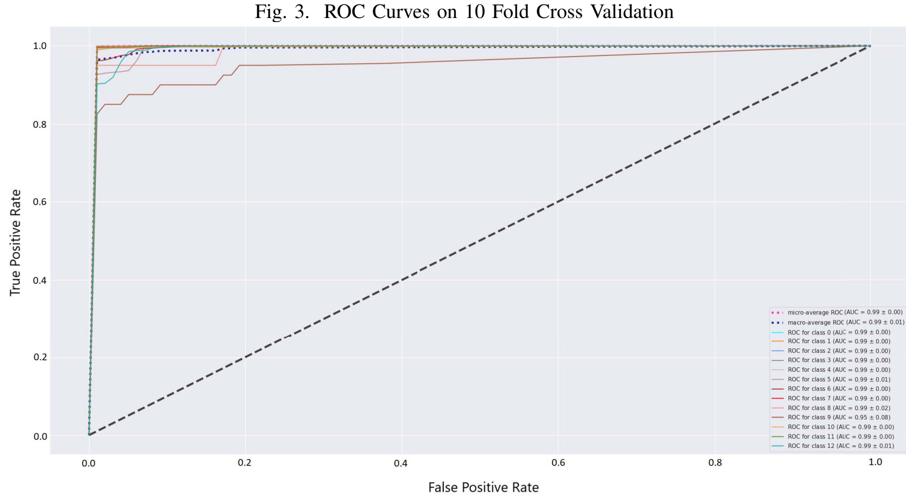



Fig. 3. ROC Curves on 10 Fold Cross Validation





[21] M. A. Ferrag and L. Maglaras, "Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," *IEEE Transactions on Engineering Management*, pp. 1–13, 2019.

[22] A. Binbusayyis and T. Vaiyapuri, "Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach," *IEEE Access*, vol. 7, pp. 106 495–106 513, 2019.

[23] B. J. Radford, B. D. Richardson, and S. E. Davis, "Sequence aggregation rules for anomaly detection in computer network traffic," *CoRR*, vol. abs/1805.03735, 2018. [Online]. Available: http://arxiv.org/abs/1805.03735

[24] A. Habibi Lashkari, G. Draper Gil, M. Mamun, and A. Ghorbani, "Characterization of tor traffic using time based features," 01 2017, pp. 253–262.

[25] ——, "Characterization of encrypted and vpn traffic using time-related features," 02 2016.

[26] M. Kuhn and K. Johnson. (2013) Applied predictive modeling. New York, NY. [Online]. Available: http://www.amazon.com/Applied-Predictive-Modeling-Max-Kuhn/dp/1461468485/

[27] J. C. Kenna, "Sir francis galton's contribution to anthropology," *The Journal of the Royal Anthropological Institute of Great Britain and Ireland*, vol. 94, no. 2, pp. 80–93, 1964. [Online]. Available: http://www.jstor.org/stable/2844375

[28] F. Galton, "Regression towards mediocrity in hereditary stature." *The Journal of the Anthropological Institute of Great Britain and Ireland*, vol. 15, pp. 246–263, 1886. [Online]. Available: http://www.jstor.org/stable/2841583

[29] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895. [Online]. Available: http://www.jstor.org/stable/115794

[30] I.-K. Yeo and R. A. Johnson, "A new family of power transformations to improve normality or symmetry," *Biometrika*, vol. 87, 12 2000.

[31] G. Box and D. Cox, "An analysis of transformations (with discussion)," *Journal of the Royal Statistical Society, Series B*, vol. 26, pp. 211–252, 01 1964.

[32] G. Batista, R. Prati, and M.-C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explorations*, vol. 6, pp. 20–29, 06 2004.

[33] G. Batista, A. Bazzan, and M.-C. Monard, "Balancing training data for automated annotation of keywords: a case study." 01 2003, pp. 10–18.

[34] N. Chawla, K. Bowyer, L. O. Hall, and W. Philip Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res. (JAIR)*, vol. 16, pp. 321–357, 01 2002.

[35] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 3, pp. 408–421, July 1972.

[36] Y. A. LeCun, L. Bottou, G. B. Orr, e. G. Müller, Klaus-Robert", G. B. Orr, and K.-R. Müller, *Efficient BackProp*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–48. [Online]. Available: https://doi.org/10.1007/978-3-642-35289-8_3

[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research - Proceedings Track*, vol. 9, pp. 249–256, 01 2010.

[38] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 315–323. [Online]. Available: http://proceedings.mlr.press/v15/glorot11a.html

[39] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.