NetMD- Network Traffic Analysis and Malware Detection

Sampath Kumar Katherasala [1] Vaddeboyina Sri Manvith [2] Ajay Therala [3] Manjari Murala [4]

[1 2 3] Tata Consultancy Services Limited, Hyderabad, India [4] IIIT Hyderabad, India

[1] sampath.katherasala@tcs.com [2] srimanvith.v@tcs.com [3] ajay.therala@tcs.com [4] muralamanjari@gmail.com

Abstract*— In this digitally connected world, data networks play a crucial role in the communication field. As there is a massive growth in data exchange, transactions and other sensitive data need to be secured. Networks must be safeguarded from malware attacks for flawless transmission of data between devices. Some of the harmful consequences of malware attacks are gaining administrative control and data breaches. Malware detection has become a significant task to get rid of those dreadful consequences and make networks secure. In this paper, we implement machine learning algorithms against the malware detection datasets NetML and CICIDS2017, and the traffic classification dataset non-vpn2016 dataset. The results are very promising and have been validated with the results obtained in the NetML Network Traffic Analytics Challenge 2020, organized by ACANETS. The overall score on the CICIDS2017 and non-vpn2016 datasets outperformed the baseline results published in the challenge, against all the five tracks (top, mid, and fine annotations).

Keywords*— Machine Learning, Malware Detection, NetML, CICIDS2017, non-vpn2016.

I. INTRODUCTION

In recent years, there has been an enormous spike in the number of electronic devices connected to the internet. With the increasing number of connected devices, malware attacks transferred over the network are also increasing regularly.

Malware is malicious software created with the intent of gaining illegal access to computers and other network devices. Malicious software can be sent over the network through web links or emails. When the user clicks the link with malicious code, these attacks run in the background and lead to a data breach. These breaches may include the loss of the victim's confidential data. There are various types of malware attacks. Malware attacks such as spyware unknowingly steal the user's information; while Ransomware encrypts the user data until the victim makes a payment. Key loggers record everything that the victim inputs to collect sensitive information about the user, such as passwords. The Botnet gains access to the system and can control the computer system remotely. Root-privilege acquisition acquires administrative rights and installs malicious applications and runs them in the background. Adware servers throw advertisements by looking at a victim's browsing history, becoming a threat to the network. It is of the utmost priority to eliminate these attacks and safeguard the devices and network from malware. In this data-driven world, machine learning (ML) and deep learning (DL) algorithms are evolving and helping to find solutions to various problems in every sector. Researchers started employing ML and DL in malware detection tasks due to the development of complex algorithms and the amount of data available on the web for analysis. These ML and DL Algorithms can be used in Network Traffic Analysis (NTA) tasks (Malware detection, traffic classification, etc.).

Researchers have employed various supervised and unsupervised learning techniques in network traffic analysis tasks. They found that supervised techniques are more effective than unsupervised techniques. Supervised techniques help in learning the patterns and predicting the class of data packets (malignant or benign).

As the dataset plays a crucial role in the performance of the ML model, the network research community needs a comprehensive, open, and up-to-date dataset for obtaining effective classification results. To address this issue, the NetML challenge introduced three datasets: NetML, CICIDS2017, and non-vpn2016. These datasets contain nearly 1.3 million labeled flows and provide researchers with a benchmarking platform to evaluate their approaches and contribute to their research on NTA.

This paper is organized into five sections. Section II provides an overview of related work on various malware detection and traffic classification problems. Section III describes the different datasets used in the challenge. Section IV explains the methodology and algorithms implemented for enhancing the results. Section V analyses the results obtained and also compares and contrasts them with the baseline results published in the challenge. Some interesting conclusions are presented in Section VI. In the appendix, we present some of the relevant exploratory data analysis of the network traffic flow features.

II. RELATED WORK

In the literature, we see many research articles published on network traffic analysis, malware detection, and also coupled with machine learning algorithms. In this paperwork, we predominantly focus on malware detection, and network traffic classification. A performance-based comparison approach was proposed by Rahim et al. [1], on the NSL-KDD dataset. They have evaluated the results based on Support Vector Machine (SVM), Random Forest (RF), and Extreme Learning Machine (ELM) algorithms on full, half, and ¼ data. They concluded that RF outperforms other approaches. Jiang et al. [2] worked on DoS and found that only fourteen classes of the CICIDS2017 dataset were considered. They compared the original CICIDS2017 features with the newly proposed features for neural networks. Ullah et Mahmoud proposed a two-level model [3]. They used a Decision tree to classify the traffic as an attack or normal at the first level and identified the type of attack at the second level using a random forest, after SMOTE-based data augmentation and edited KNN. Their

two-level method has been tested on the UNSW-NB15 and CICIDS2017 datasets. A two-step approach was proposed by Ustebay et al. [4]. on the CICIDS2017 dataset. The most useful features were identified using the Recursive Feature Elimination (RFE) technique, and these features are used for training the neural network model. The models developed do not perform well when all types of attacks are considered. Otherwise, performance results are not that great.

Arnaud et al. [5] have worked on the MLP algorithm and have built a golden model. They used neural networks to evaluate the results on the CICIDS2017 dataset. Their approach provided better results on the complete dataset and good performance on training the model without the IP addresses and destination port features. A new classification model called Arc margin was developed by Xiaojun Wang et al. [6], which closely maps network traffic samples from the same category. They experimented on three datasets: non-vpn2016, CICIDS2017, and CICIDS2012, obtaining precision and recall values of 0.9857, 0.9853 on non-vpn2016, 0.9934, and 0.9933 on CICIDS2017, and 0.9971, and 0.9971 on the CICIDS2012 dataset.

A model based on LSTM and CNN for network traffic classification was proposed by Feifei Hu et al. [7] on the non-vpn dataset. They have obtained a precision, recall, and f1 score of 97.4, 97.5, and 96.8 respectively on the non-vpn dataset. M. Lopez-Martin et al. [8] have experimented and shown that usage of a single deep learning technique (such as CNN or RNN) compared to combinations of techniques such as (CNN+LSTM, RNN+LSTM) has more advantages in classifying network traffic.

Onur Barut et al. [9] have performed research on the importance of NTA in application classification and malware detection. They have generated three datasets, namely, NetML, CICIDS2017, and non-vpn2016, and implemented several machine learning algorithms like Random Forest, SVM, and MLP on these datasets. They have presented challenge baseline results on seven different tracks. However, they did not perform data balancing techniques. There is a scope to improve the performance of proposed ML models.

All of these related works have motivated us, and we were able to outperform the NetML Network Traffic Analytics Challenge 2020 baseline results and leaderboard participants organised by the ACANETS challenge [12].

III. DATASETS

There have been a plethora of research attempts to analyze and classify network traffic using a variety of datasets. Nevertheless, with the open datasets we have in computer vision research, such as ImageNet and COCO, it is very difficult to find a comprehensive dataset for researchers in networking. However, in the recent work [9], to enable data-driven machine learning-based network flow analytics, they introduced a benchmark traffic dataset, known as NetML, curated from open sources for malware detection and network traffic classification. They have released the traffic flow features and different levels of annotations, aiming to present a common dataset for the research community. This dataset consists of three different sets in which two data samples are prepared for malware detection, NetML and CICIDS 2017, and one dataset is for traffic classification, vpn2016.

Malware Detection Datasets: NetML & CICIDS2017

NetML and CICIDS2017 are the two datasets created with the raw traffic captured from the Stratosphere IPS [10] website and the Canadian Institute of Cybersecurity (CIC) [11], respectively. These datasets are captured for detecting malware. Both datasets were further divided into top and fine-grained annotations. In the top-level annotation of NetML [9] and CICIDS2017 [9] datasets, captured traffic data is classified as benign or malware, as shown in Table I and Table II. At the top level, if a packet is classified as malware, then in fine-grained annotation the type of malware is classified. In the NetML dataset, twenty different malware classes are there, such as Dridex, Trickster, Ursnif, etc., as shown in Table III. In the CICIDS 2017 dataset, there are seven different types of malware classes, such as portScan, DoS, infiltration, etc., that are classified in the fine-grained annotation as shown in Table IV.

TABLE I. CLASS DISTRIBUTION OF NETML - TOP DATASET

Class	***Number of Samples***
benign	311273
malware	75995

TABLE II. CLASS DISTRIBUTION OF NETML – FINE-GRAINED DATASET

Class	***Number of Samples***
benign	242661
malware	198455

TABLE III. CLASS DISTRIBUTION OF CICIDS2017 - TOP DATASET

Class	**Adload**	**Artemis**	**Downware**	**CCleaner**	**Cobalt**
Samples	75995	57796	30442	37271	31458
Class	**PUA**	**Dridex**	**Emotet**	**HTBot**	
Samples	8238	18627	15767	15289	
Class	**TrickBot**	**Trickster**	**Ramnit**	**Sality**	**Tinba**
Samples	4074	4020	8139	6162	4732
Class	**BitCoinMiner**		**Trojan Downloader**		**Miner Trojan**
Samples	45907		3849		8482
Class	**MagicHound**		**WebCompanion**	**Ursnif**	**benign**
Samples	9208		400	1379	33

TABLE IV. CLASS DISTRIBUTION OF CICIDS2017 - FINE DATASET

Class	**DDoS**	**DoS**	**ftp-patator**	**infiltration**
Samples	198455	122430	36136	23806
Class	**portScan**	**ssh-patator**	**webAttack**	**benign**
Samples	3168	1972	1617	53532

Traffic Classification Dataset: non-vpn2016

The non-vpn2016 dataset is collected from NetML Network Traffic Analytics Challenge 2020 organized by ACANETS [12]. This dataset primarily emphasizes application

classification. Three levels of annotation are assigned to this dataset in which top-level annotation groups the captured traffic data into seven classes, namely P2P, chat, audio, email, file_transfer, tor, and video, as shown in Table V. Mid-level annotation consists of eighteen different applications (Gmail, Google, Netflix, Skype, Youtube, Facebook, etc.,) as shown in Table VI. Fine-level annotation of thirty-one low–level classes in an application (skype_video, facebook_audio, tor_google, tor_twitter, etc.,) as shown in Table VII. In the non-vpn2016, dataset four sets of features were extracted. They are Meta Data Features, TLS Features, DNS Features, and HTTP Features. Protocol-specific features are extracted only if the flow contains packets with any one of the protocols, whereas metadata features are extracted for any kind of flow.

TABLE V. CLASS DISTRIBUTION OF NON-VPN2016 - TOP DATASET

Class	**P2P**	**audio**	**chat**	**email**
Samples	992	121930	6772	4300
Class	**file_transfer**	**tor**	**video**	
Samples	25896	128	3693	

TABLE VI. CLASS DISTRIBUTION OF NON-VPN2016 - MID DATASET

Class	**aim**	**email**	**facebook**	**ftps**	**gmail**
Samples	1011	12477	106596	1995	1089
Class	**google**	**hangouts**	**icq**	**netflix**	**scp**
Samples	9	113940	1056	822	450
Class	**sftp**	**skype**	**spotify**	**torrent**	**twitter**
Samples	453	140133	546	2496	12
Class	**vimeo**	**voipbuster**	**youtube**		
Samples	1095	7047	1968		

IV. METHODOLOGY

In this work, we have divided our methodology into preprocessing and classification of algorithms. Data preprocessing is the process of applying transformations on raw data to clean the data. We apply various preprocessing techniques, such as scaling and balancing. We have scaled the data using the Standard Scaler. The datasets we experimented on have a high-class imbalance, as shown in Fig. [a, c, e, g, i, k, l]. Machine learning models may suffer from bias due to unbalanced data. Hence, to avoid bias problems, we have also experimented with balancing techniques as shown in Fig. [b, d, f, h, j]. These help in either upsampling or downsampling the number of samples in each class such that each class contains an equal number of samples. Techniques such as Undersampling, Oversampling, and SMOTE were experimented with. Of these, SMOTE gave better results.

In the next part, we have employed various Machine Learning algorithms; Random Forest, Support Vector Machine, Logistic Regression, Naïve Bayes, Adaboost, CatBoost, and XGBoost on

- unscaled and unbalanced dataset
- unscaled and balanced dataset
- scaled and unbalanced dataset
- scaled and balanced dataset

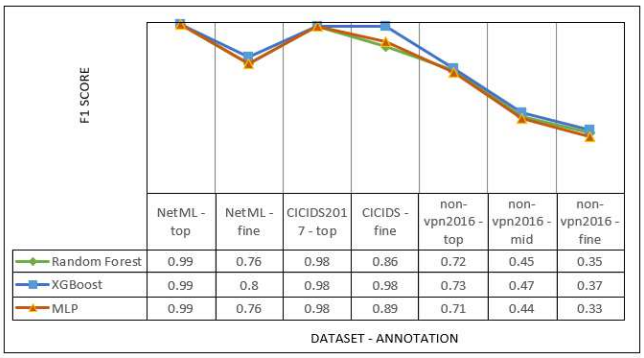
Out of all these, the performance results are good against scaled and balanced datasets.

TABLE VII. CLASS DISTRIBUTION OF NON-VPN2016 - FINE DATASET

Class	**aim_chat**	**email**	**facebook_audio**
Samples	346	4372	63349
Class	**gmail_chat**	**hangouts_audio**	**hangouts_chat**
Samples	415	38201	366
Class	**ftps_up**	**skype_chat**	**scp_up**
Samples	176	4880	84
Class	**scp_down**	**sftp_down**	**sftp_up**
Samples	89	98	67
Class	**facebook_chat**	**skype_audio**	**facebook_video**
Samples	433	17149	357
Class	**icq_chat**	**hangouts_video**	**ftps_down**
Samples	353	1231	535
Class	**netflix**	**youtube**	**voipbuster**
Samples	348	735	2754
Class	**vimeo**	**torrent**	**tor_youtube**
Samples	437	1016	104
Class	**tor_vimeo**	**tor_twitter**	**tor_google**
Samples	17	4	3
Class	**tor_facebook**	**spotify**	**skype_video**
Samples	3	207	584

V. RESULTS

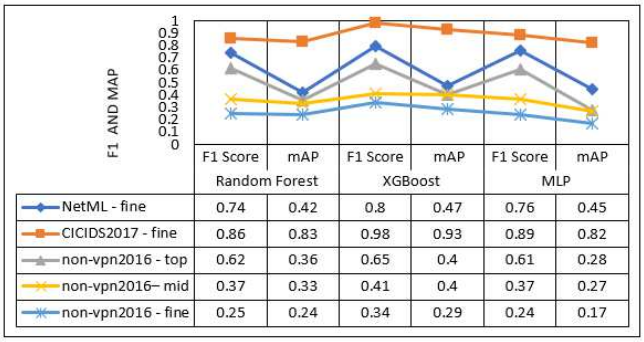
The NetML and CICIDS2017 datasets are used for malware detection problems, while the nonvpn2016 dataset is used for traffic classification problems. For all the datasets, we have calculated validation accuracy based on 20% validation data as shown in Figure 1. Top Annotations of both NetML and CICIDS2017 datasets are for binary classification problems. Hence, we have calculated the True Positive Rate and False Alarm Rate for assessing the performance of the model. Fine annotation of the NetML and CICIDS2017 datasets, the non-vpn2016 dataset is used for traffic classification purposes, and the performance metrics F1 score and mAP score are evaluated for assessing the model performance.



	NetML - top	NetML - fine	CICIDS2017 - top	CICIDS - fine	non-vpn2016 - top	non-vpn2016 - mid	non-vpn2016 - fine
Random Forest	0.99	0.76	0.98	0.86	0.72	0.45	0.35
XGBoost	0.99	0.8	0.98	0.98	0.73	0.47	0.37
MLP	0.99	0.76	0.98	0.89	0.71	0.44	0.33

Fig. 1. validation accuracies on 20% dataset

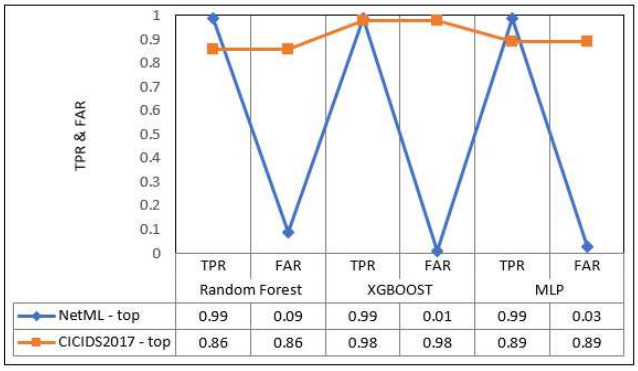
For all the annotations of the three datasets, XGBOOST performed better than Random Forest and MLP.



	F1 Score (Random Forest)	mAP (Random Forest)	F1 Score (XGBoost)	mAP (XGBoost)	F1 Score (MLP)	mAP (MLP)
NetML - fine	0.74	0.42	0.8	0.47	0.76	0.45
CICIDS2017 - fine	0.86	0.83	0.98	0.93	0.89	0.82
non-vpn2016 - top	0.62	0.36	0.65	0.4	0.61	0.28
non-vpn2016– mid	0.37	0.33	0.41	0.4	0.37	0.27
non-vpn2016 - fine	0.25	0.24	0.34	0.29	0.24	0.17

Fig. 2. F1 score and Mean Absolute Precision

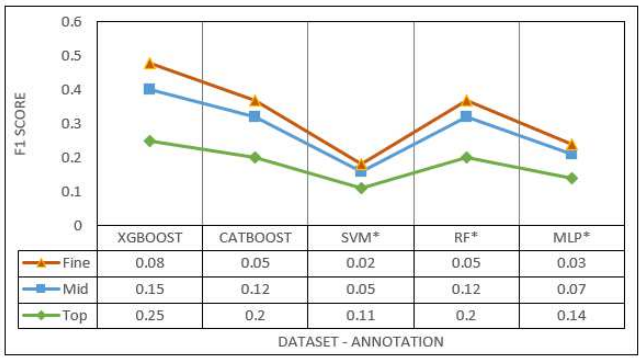
As shown in Figure 2, the F1 score and mAP values of the fine annotation of NetML are 0.80 and 0.47, respectively, and those of CICIDS2017 are 0.98 and 0.93 respectively. The F1 score and mAP values of the top, mid, and fine annotations of the non-vpn2016 dataset are 0.65 and 0.40; 0.41 and 0.40; 0.34 and 0.29 respectively. XGBOOST performed better than Random Forest and MLP, for both F1 and mAP.



	TPR	FAR	TPR	FAR	TPR	FAR
	Random Forest		XGBOOST		MLP	
NetML - top	0.99	0.09	0.99	0.01	0.99	0.03
CICIDS2017 - top	0.86	0.86	0.98	0.98	0.89	0.89

Fig. 3. TPR and FAR scores/values

In Figure 3, The TPR and FAR values of the top annotation of NetML are 0.99 and 0.01 respectively, and those of CICIDS2017 are 0.98 and 0.98 respectively. XGBOOST performed better than Random Forest and MLP on TPR and FAR.

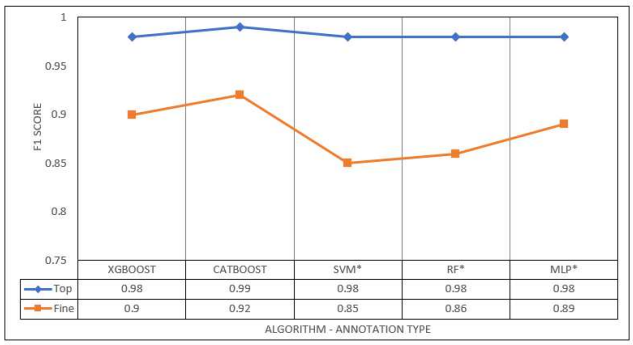


	XGBOOST	CATBOOST	SVM*	RF*	MLP*
Fine	0.08	0.05	0.02	0.05	0.03
Mid	0.15	0.12	0.05	0.12	0.07
Top	0.25	0.2	0.11	0.2	0.14

DATASET - ANNOTATION

Fig. 4. Challenge Scores of the non-vpn2016 dataset

As shown in Figure 4, for all the annotations of the non-vpn2016 dataset, XGBOOST (top - 0.25, mid – 0.15, fine – 0.08) performed better than other algorithms. Our experimental results outperformed the ACANETS baseline results, as shown in Fig. [o, p, q].



	XGBOOST	CATBOOST	SVM*	RF*	MLP*
Top	0.98	0.99	0.98	0.98	0.98
Fine	0.9	0.92	0.85	0.86	0.89

ALGORITHM - ANNOTATION TYPE

Fig. 5. Challenge Scores of the CICIDS2017 dataset

From Figure 5, it is emphasized that for all the annotations of the CICIDS2017 dataset, CATBOOST (top – 0.99, fine – 0.92) performed better than other algorithms. Our experimental results outperformed the ACANETS baseline results, as shown in Fig. [m, n].

VI. Conclusion

Secure networks help in defending against malware attacks and safeguarding computer systems from data breaches. Detecting malware is required to protect the network from malicious activities and make networks secure. In this paper, we have performed malware detection on NetML, CICIDS2017, and non-vpn2016 datasets by applying machine learning algorithms. In this experiment, we applied various machine learning algorithms such as Random Forest, SVM, Naïve Bayes, XGBoost, CatBoost, and MLP. We have published our results in the NetML Network Traffic Analytics Challenge 2020, organized by ACANETS. The results are outperformed against the baseline results on the challenge leaderboard. Metrics calculated for evaluation of our work include FAR and TPR for binary classification, mAP, and F1 score for multi-class classification. As per the performance analysis, we found that XGBoost performed well for all annotations on the non-vpn2016 and NetML datasets. CatBoost performed well on CICIDS2017 – top and MLP performed well on fine annotation.

References

[1] Ahmad, M. Basheri, M. J. Iqbal and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," in IEEE Access, vol. 6, pp. 33789-33795, 2018, DOI: 10.1109/ACCESS.2018.2841987.

[2] J. Jiang et al., "ALDD: A Hybrid Traffic-User Behavior Detection Method for Application Layer DDoS," 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018, pp. 1565-1569, DOI: 10.1109/TrustCom/BigDataSE.2018.00225.

[3] I. Ullah and Q. H. Mahmoud, "A Two-Level Hybrid Model for Anomalous Activity Detection in IoT Networks," 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2019, pp. 1-6, DOI: 10.1109/CCNC.2019.8651782.

[4] S. Ustebay, Z. Turgut and M. A. Aydin, "Intrusion Detection System with Recursive Feature Elimination by Using Random Forest and Deep Learning Classifier," 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), 2018, pp. 71-76, DOI: 10.1109/IBIGDELFT.2018.8625318.

[5] Rosay, Arnaud & Carlier, Florent & Leroux, Pascal. (2020). MLP4NIDS: An Efficient MLP-Based Network Intrusion Detection for CICIDS2017 Dataset, DOI:10.1007/978-3-030-45778-5_16.

[6] Mo, Chen & Xiaojuan, Wang & Mingshu, He & Lei, Jin & Javeed, Khalid & Wang, Xiaojun. (2020). A Network Traffic Classification Model Based on Metric Learning. Computers, Materials & Continua, DOI:10.32604/cmc.2020.09802.

[7] Hu, Feifei & Zhang, Situo & Lin, Xubin & Wu, Liu & Liao, Niandong & Song, Yanqi. (2021). Network Traffic Classification Model Based on Attention Mechanism and Spatiotemporal Features. DOI: 10.21203/rs.3.rs-353938/v1.

[8] Lopez-Martin, Manuel & Carro, Belén & Sanchez-Esguevillas, Antonio & Lloret, Jaime. (2017). Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things. IEEE Access. DOI:10.1109/ACCESS.2017.2747560.

[9] NetML: A Challenge for Network Traffic Analysis https://arxiv.org/abs/2004.13006

[10] Stratosphere. 2015. Stratosphere Laboratory Datasets. (2015). https://www.stratosphereips.org/datasets-overview [Online; accessed 12-March-2020].

[11] Iman Sharafaldin, Arash Habibi Lashkari, and Ali Ghorbani. 2018. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. 108–116.

[12] NetML Network Traffic Analytics Challenge 2020: https://eval.ai/web/challenges/challenge-page/526/overview : Leaderboard Results: non-vpn2016 Dataset : Top annotation : https://eval.ai/web/challenges/challenge-page/526/leaderboard/1471 ; Mid annotation: https://eval.ai/web/challenges/challenge-page/526/leaderboard/1472 ; Fine annotation https://eval.ai/web/challenges/challenge-page/526/leaderboard/1473 ;

CICIDS2017 Dataset : Top annotation: https://eval.ai/web/challenges/challenge-page/526/leaderboard/1474; Fine annotation : https://eval.ai/web/challenges/challenge-page/526/leaderboard/1475

APPENDIX

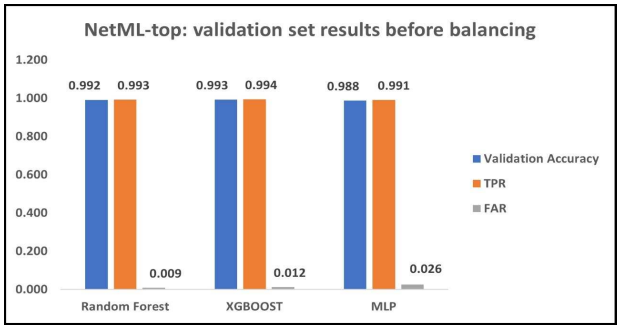


Fig. a. NetML-top: validation set results before balancing

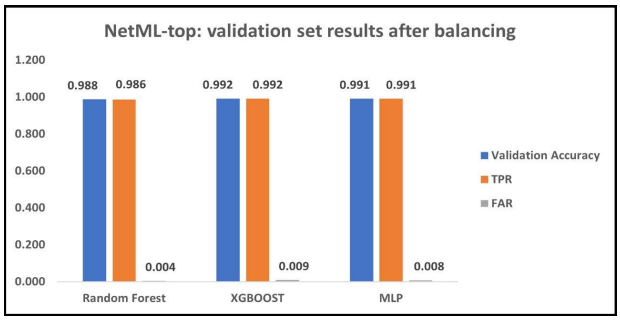


Fig. b. NetML-top: validation set results after balancing

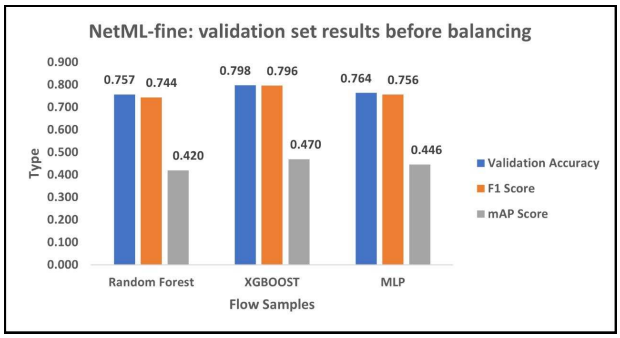


Fig. c. NetML-top: validation set results before balancing

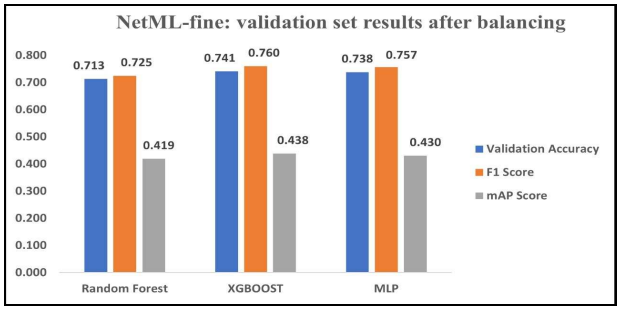


Fig. d. NetML-fine: validation set results after balancing

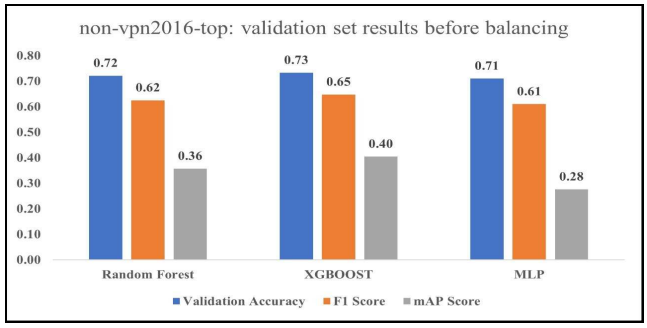


Fig. e. non-vpn2016-top: validation set results before balancing

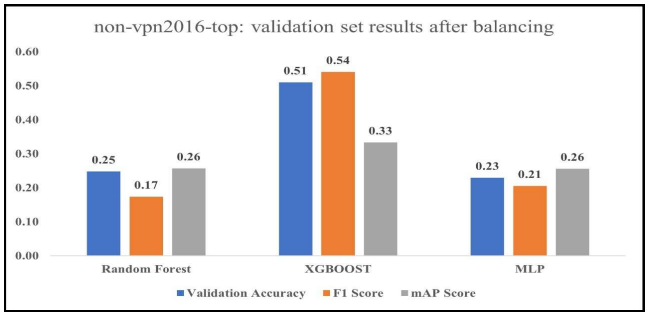


Fig. f.non-vpn2016-top: validation set results after balancing

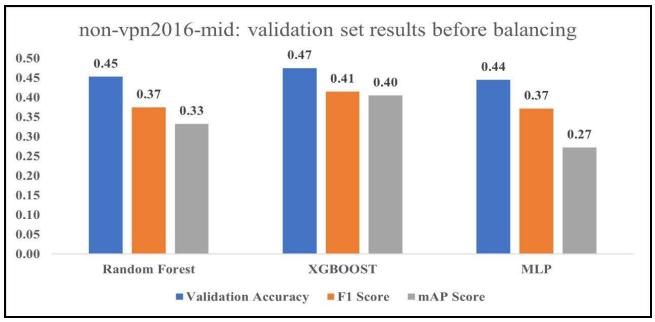


Fig. g. non-vpn2016-mid: validation set results before balancing

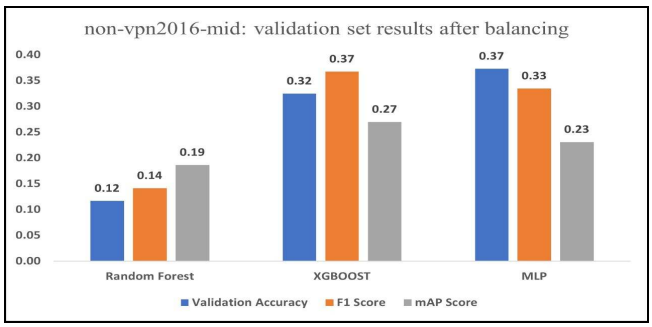


Fig. h. non-vpn2016-mid: validation set results after balancing

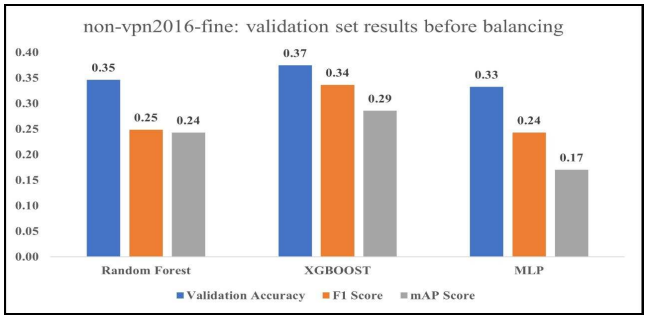


Fig. i. non-vpn2016-fine: validation set results before balancing

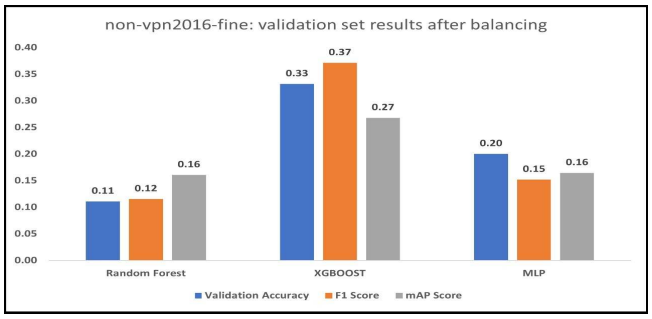


Fig. j. non-vpn2016-fine: validation set results after balancing

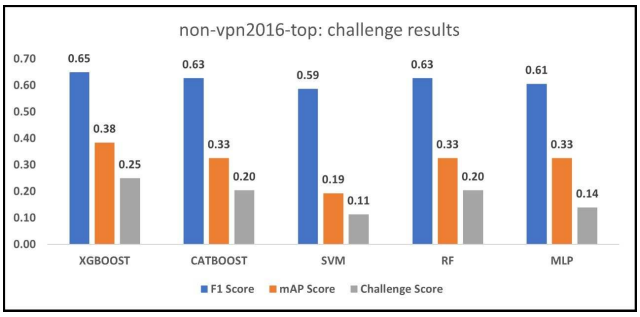


Fig. o. non-vpn2016-top: challenge results

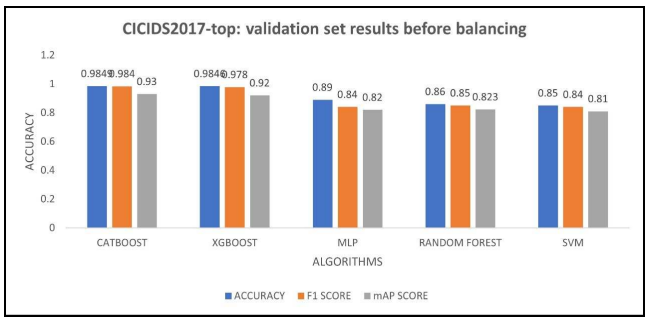


Fig. k. CICIDS2017-top: validation set results before balancing

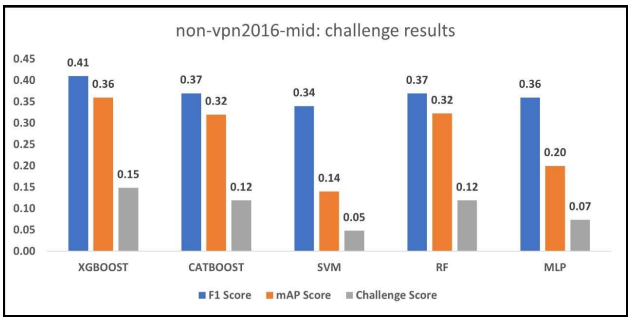


Fig. p. non-vpn2016-mid: challenge results

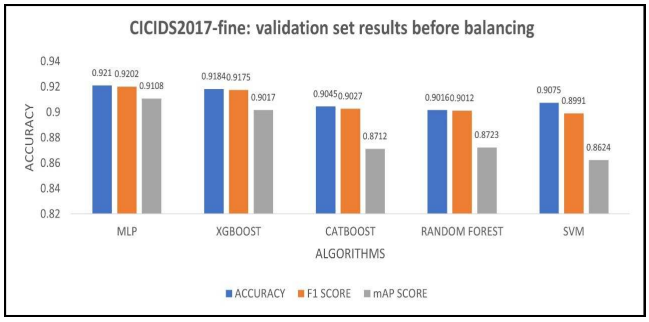


Fig. l. CICIDS2017-fine: validation set results before balancing

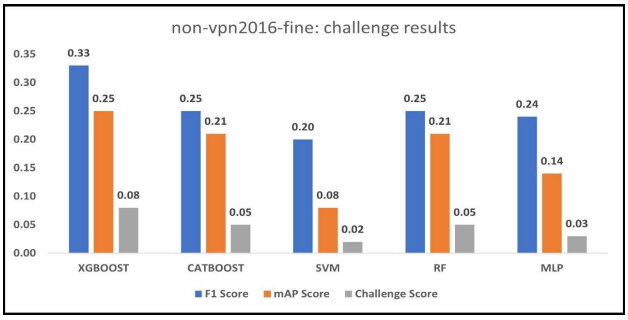


Fig. q. non-vpn2016-fine: challenge results

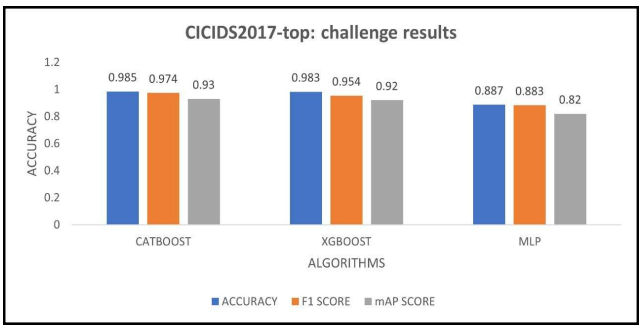


Fig. m. CICIDS2017-top: validation set results after balancing

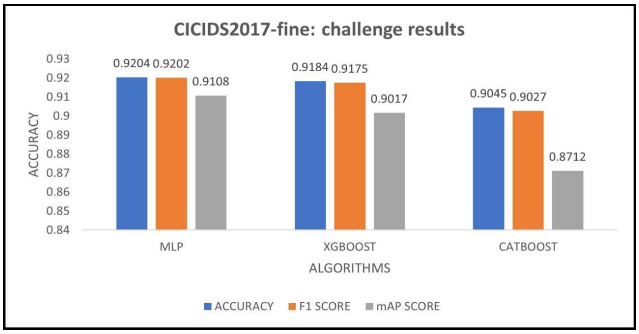


Fig. n. CICIDS2017-top: validation set results after balancing