Dicle University
Journal of Engineering
https://dergipark.org.tr/tr/pub/dumf
duje.dicle.edu.tr

Research Article

FastTrafficAnalyzer: An Efficient Method for Intrusion Detection Systems to Analyze Network Traffic

Recep Sinan ARSLAN[1*]

[1] Kayseri University, Department of Computer Engineering, sinanarslanemail@gmail.com, Orcid No: 0000-0002-3028-0416

ARTICLE INFO

Article history:

Received 9 May 2021
Received in revised form 2 September 2021
Accepted 7 September 2021
Available online 27 September 2021

Keywords:

Network intrusion detection, machine learning, feature conversion, cyber security

Doi: 10.24012/dumf.1001881

* Corresponding author

ABSTRACT

Network intrusion detection systems are software or devices used to detect malignant attackers in modern internet networks. The success of these systems depends on the performance of the algorithm and method used to catch attacks and the time it takes for it. Due to the continuous internet traffic, these systems are expected to detect attacks in real time. In this study, using a proposed pre-processing, internet traffic data becomes more easily processable and traffic is classified by network analysis with machine learning techniques. In this way, the traffic analysis time was significantly shortened and a high level of success was achieved. The proposed model has been tested in the CSE-CIC-IDS2018 dataset and its advantaged verified. Experimental results i) 99.0% detection rate was achieved in the ExtraTree algorithm for binary classification, while a reduction of 82.96% was achieved in the processing time per sample; ii) For multiclass (15 class) detection, 98.5% detection rate was achieved with the Random Forest algorithm, while a 64.43% shortening was achieved in the processing time per sample. As a result, similar classification rate with the studies in the literature has been achieved with much shorter test time.

Introduction

With the emergence and development of network structures, almost all institutions, organizations and companies have established their own large-scale LANs. Large amount of network and sensitive information are stored in these networks [13]. In addition, new device types and network structures have emerged, such as 5G, cloud computing, and the Internet of Things (IoT). With the growth of these systems and networks, ensuring their cyber security is critical [14]. According to the 2021 Cyber Threat Defense Report (CyberEdge, 2021) [4] released by the CyberEdge team and compiled from data provided by 1200 IT security experts working in 19 different sectors in 17 different countries, attacks to companies connected to the internet network have been increased (Figure-1).

These threats in recent years contain denial of service, malicious software attacks, spyware attacks, and continues threats. Especially, advanced persistent threats are dangerous and costly. Because they are carried out by experts in the field and are mostly supported by the governments. They are long-term attacks against governments or big companies for the purpose of data stealing and sabotage of the infrastructure. According

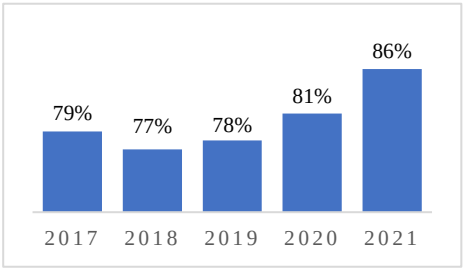

Figure 1. The number of attacks in the last 5 years (CyberEdge, 2021).

to the M-trends 2021 report, as of September 2020, the average waiting time (dwell time), which indicated that attackers have been working on a system or network for more than one year without detection, is 416 days, while the median waiting time between October 1, 2019 and September 30, 2020 is only it has decreased to 24 days [5].

First place of defense in ensuring cyber security in computer networks consist of a security infrastructure that includes Access control, privacy management, authentication and various security policies. However, despite all this defensive structure, it can be targeted by cyber attackers for reasons such as operations problems, other zero-day

exploits and various systemic vulnerabilities. As the second line of defense behind the firewall, IDS systems are tasked with accurately identifying malicious network attacks, providing real-time monitoring, dynamic protection measures, and creating strategies [6]. An IDS can be placed on all independent distributed hosts in a network, or they can be located on a central dedicated server and distribute its service over the network. Network intrusion detection systems (NIDS) designed to detect attacks from a single center for all systems on a network. NIDS monitor network information such as network traffic, protocols, flow metadata, activity logs and try to detect abnormal events. The main purpose of an IDS system is to detect unauthorized computer use and malicious network traffic that is not possible when using a traditional firewall. This system makes it highly protected against malicious acts that endanger the usability, integrity or privacy of computer systems [7].

IDS systems are basically divided into two types such as signature-based intrusion detection systems (SIDS) and anomaly-based systems (AIDS). SIDS uses patterns in its database to detect an attack, previously prepared for a known attack type. When a pattern matches with the signatures in the databases, an alarm is generated. For detection, the host's logs and command sets are analyzed. On the other hand, AIDS is an approach that can produce solutions to many constraints of signature-based systems. In this method, the normal behavior of a computer or network is extracted with the help of machine learning. When there is a deviation between an observed suspicious and the normal behavior pattern of the systems, it is interpreted as an attack. This technique is based on the fact that malicious attackers have different behavioral patterns from normal users [8].

Machine learning approach to attack detection has been studies by researchers for a long time [9, 30]. Large volumes of network metadata and security-related data allow ML to be applied to detect intrusion into systems. Cisco Stealthwatch [10], Darktrace [11], Vectra Cognition [12] which are modern commercial IDSs use ML as part of their security strategy.

In the light of the challenges and requirements given above, a machine learning-based model is proposed for the classification of suspicious internet traffic. The model was evaluated with CSE-CIC-IDS2018 dataset with both multiple and binary classification scenarios.

(1) A new pre-processing model is proposed, which provides significant gains in the training and testing time of the classification models. Thus, it was possible to achieve high classification results in a much shorter time compared to similar studies.

(2) Tests have been carried out with 9 different ML techniques in order to observe the effect of this proposed new pre-processing step on both the reduction rate in the processing time and the classification results, and the results are given comparatively.

(3) Repeated experimental results under similar conditions have been shown that this proposed approach provides similar classification results with existing current studies, while achieving these results 82.96% faster than them. Thus, a model that will contribute to the real-time operation requirements of IDS systems is proposed.

The remainder of the article is organized as follows. In Section 2, the current studies on CSE-CIC-IDS2018 dataset are mentioned in detail. Originally designed pre-processing model are given in Section-3. Experimental results are presented in Section-4. The study has been concluded in Section-6.

Related Works

Intrusion detection systems (IDS) are an up-to-date subject that is studies both in the field of cyber security and as an academic research. For this reason, many studies have been carried both in this area in recent years. Since the proposed model was tested with the IDS2018 dataset, which is one of the best examples of current internet traffic, in this section, studies only conducted with the same dataset are given in this section.

In [17] used the SMOTE datasets balancing algorithm to oversampling the samples of minority classes and then evaluated the results with 6 different ML algorithms (RF, DT, KNN, Adaboost, GB and LDA). The experimental results indicated that the proposed model performed better that the existing models.

In the study [18] on the same dataset, a GMM-based approach is proposed to solve the sample imbalance problem between classes to increase classification performance rate. Thus, by using SMOTE and GMM together, the number of samples in classes was balanced and the results were evaluated. Accordingly, a sample CNN model was designed and demonstrated that balancing the classes improves the learning process and contributes directly to performance enhancement.

In the study conducted on the use of artificial intelligence techniques in IDS systems, it is aimed to classify the botnet attack, which is a serious threat to the finance and banking sectors [19]. In tests performed in IDS 2018 dataset with a designed ANN, a low FP rate of 3% was achieved. ANN have been shown to produce good results in detecting botnet attacks.

In [20], a design has been made in which deep neural network and deep autoencoders are included in the learning process together. A model that uses grid search and random search algorithms together is proposed to determine optimum model parameters. Deep autoencoder, autoencoder and stack autoencoder are used for feature extraction. As a result, more successful results have been obtained in multiclass classification than previous approaches.

An artificial neural network-based model has been proposed [21]. It is aimed at ensuring security in software defined networks, in which packet analysis is separated from the traffic routing processes. It has been shown in the

tests that it is more beneficial to used much more complex and multilayer neural network to increase classification performance. It has been stated that LSTM-based models produce more successful results for real life usable system designs.

Increasing data flow in the network can cause attacks and intrusions. In order to prevent attacks, it is necessary to quickly classify large amounts of data with little cost. In [22], unsupervised feature selection method is proposed to avoid the cost of tagging network traffic. Thus, it is aimed to achieve better classification results while reducing computational complexity. Experimental results with IDS 2018 have confirmed that the proposed method is suitable for use in LAN and mobile ad-hoc networks and other networks with variable data density.

Based on the insufficiency of signature-based approaches in the detection of cyber-attacks in today's conditions, a deep learning-based approach has been proposed [23]. It is a working model only for DoS attacks. For this reason, tests were carried out on the records for DoS among the types of attacks found in the KDD and IDS2018 dataset. As a result of the tests performed with CNN and RNN-based models, it has been shown that the CNN model is more successful.

Apart from the ones given in this section, many survey articles covering all IDS related studies from past to present have been prepared [24-26], and they have been evaluated comparatively regardless of the data set.

System Architecture

The evaluation of the proposed model in dataset was prepared in Python and tested with a structure supported by pandas [1], sklearn [2] and xgboost [3] modules. In the current application, 9 supervised machine learning classifiers were used. 6 of them are tree-based, decision tree, random forest, gradient-boosted trees (gradient boost), random decision trees (extra tree) and extreme gradient boosted trees(xgboost), 1 of them is neighborhood-based, k-nearest neighbor (KNN) and 2 of them are SVM based, linear SVC and logistic regression.

The NIDS problem

The main purpose of NIDS is to follow the system operating process and behavior. In this way, they determine when the attacks occur and create warning messages when necessary. Thus, it is to ensure that field experts can mitigate the bad consequences that will arise. As an input to an IDS system, traffic statistics in a computer network, packet headers and information to be obtained from packet content and process behavior, system call traces, application logs, file system change information coming from host computers can be given. And its output can be like a tagging for each input, attack scoring or categorization. Tags can be both benign or malicious, as well as multi-class attacks such as brute-force, infiltration, SQL injection. Using machine learning algorithms, this problem can be modelled as a classification or anomaly-based detection problem. In the training of the model to be prepared for anomaly detection, a data set consisting of information obtained from the network and hosts will be sufficient. On the other hand, for classification, tagged inputs is required. Thus, the system that emerged after completing the model training, can be deployed to detect new attacks.

System design and proposed model

A model has been designed in order to observe the effect of the proposed model on the classification results in IDS systems and the time required to obtain these results. The flow-chart of designed model is shown in Figure-2. The CIC-IDS 2018 dataset was used to evaluate the study. Since the experiments are carried out for both multiple and binary classification, the dataset has been subjected to a separate process to used two classification model. Two test processes were carried out, one with the original dataset and the other with the dataset subjected to proposed pre-processing.

In the two test processes carried out for comparison, feature selection, under-sampling, random separation for training and testing, standardization and classification models were kept exactly the same. The purpose of this is to observe the difference and contribution of the proposed model from traditional approaches.

In the feature selection, 80 features found in the original dataset were reduced to 42 features. In this way, both the processing time was saved and the noisy features that negatively affected the training of the model were removed. In addition, features that cannot be mathematically processed such as IP address are eliminated in this section. In the next step, under-sampling was applied. The reason for this is that there are over 18M data in dataset and it is very difficult to transform and process such large amounts of data into vectors for KNN or SVM. For this reason, the number of data has been reduced with under-sampling and a subset that best represents the original dataset is obtained. Then, this data was separated randomly to be used in the training and testing as 70%-30%. Finally, after a standardization of the data, both feature vectors emerging are given to the classification model. The results were compared in terms of both the classification performance rate and the processing time.

When the dataset is examined, it is seen that both the complete parts and the fraction parts of the values for each feature increases up to 10 digits (12.00215875 or 8214251652.0012). Working with high-digit numbers for 40 different properties can cause problems, especially in terms of processing time. This required processing time is quite long for most real-time IDS systems, and it is not directly possible to use the developed models in real life. In order to find a solution to this problem, all data were evaluated according to the approach shown in Figure-3 and all values were changes to 0 or 1.

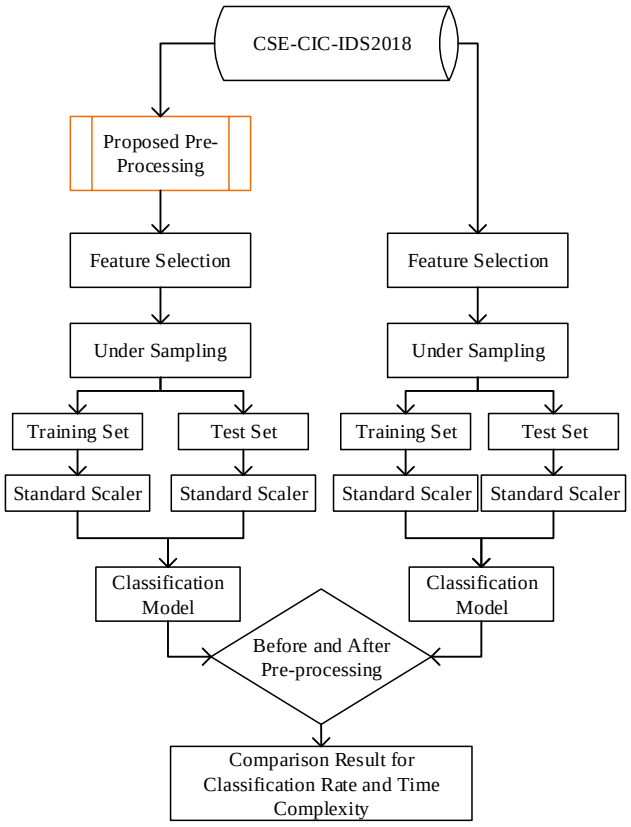

Figure 2. System overview of proposed model.

Regarding the proposed model, first of all, the average value in the whole dataset was calculated separately for each feature. According to this average, the value of each feature is changes to 1 if it is greater than the average value and 0 if it is less than the average value. Thus, the average value calculated on a feature basis is used as a threshold. Thus, instead of working with decimal numbers of internet traffic, it is possible to work with feature vectors with values of 0 and 1. As a result, it is aimed to produce feature vectors that will be easier and faster to work with in both training and testing processes of classification algorithms.

As the data is changes with the approach shown in Figure-3, it is necessary to observe whether the emerging new feature vector eliminates meaningful data. If significant value is lost, classification performance will decrease. In this case, even if you have a faster classification and testing process, performance will be lost. In order to observe this situation, the experiments, the details of which are given in the next section, have been carried out.

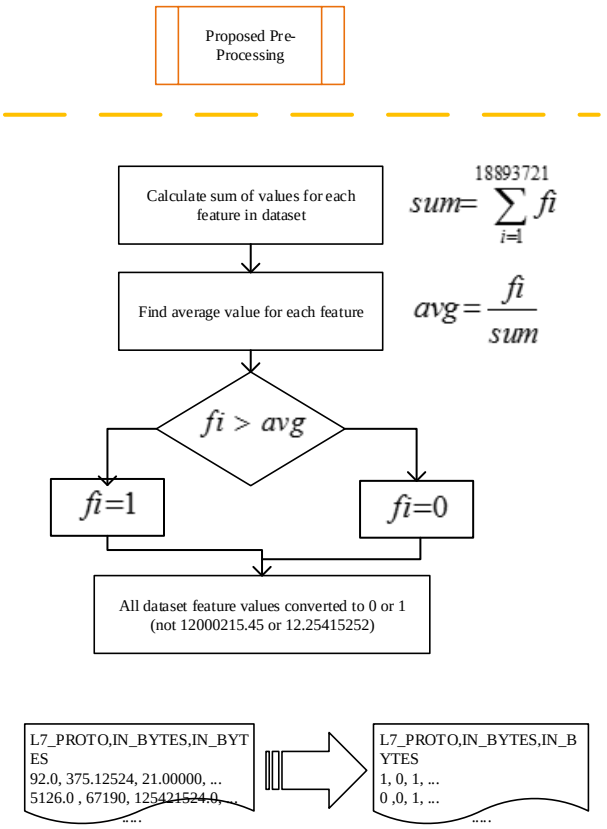

Figure 3. Details of proposed pre-processing model.

Experimental Design and Results

In this study, it is tried to show that contribution of new pre-processing method in the detection of attacks in IDS systems. Experimental environment, dataset details and test results, which are performed in multiclass and binary, are given in this section and the results are discussed with similar studies.

Experimental design

The test process was carried out on Intel Xeon a 3.5 GHz 8 core processor with 32 GB Ram and Ubuntu 18.04 operating system. It was developed in Python programming language by using sklearn library and tensorflow for machine learning. Imbalanced-learning library is used for under-sampling.

CSE-CIC-IDS2018 dataset

CSE-CIC-IDS2018 was prepared by The Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC) for use in attack detection and prevention. It is a dataset representing the traffic flow and attack types of real life network structure and is distributed as open source [15]. It consists of 80 features, including tags. Statistical information about internet traffic constitutes most of the features and obtained with the CICFlowMeter and outputted and based on the flow of network traffic. The details of the scenario and the process of extracting traffic characteristics are explained in [15,16].

Almost all of the features in the dataset consist of continuous features such as dst_ports, protocols,

timestamps, labels. It includes 14 types of attacks in 6 different scenarios including Brute-Force, Botnet, Web attacks, DDos and infiltration. The total number of records is 18893270. Of these, 16635579 records (%83.07%) represent benign traffic, while the remainder represent malicious. In this study 9.73% is used for training and 4.17% for testing. The amount of data to be used in the training and testing phase of the proposed model is shown in Table-1. It is aimed to prove the objectivity of the proposed model and that is a viable method in real systems by performing the tests with CSE-CIC dataset which is quite large, contains different attack types and internet traffic.

Table 1. CSE-CIC-2018 data distribution (before and after random under sampling with majority).

Category	Attack Types	Size - %		Train – Test (%70-%30) (After Under Sampling)		
Benign		16635579	88.050	442	190	0.028%
DDoS	HOIC	1080858	5.721	756601	324257	47.852%
	LOIC-http	307300	1.627	215110	92190	13.605%
	LOIC-UDP	2112	0.011	1478	634	0.094%
Botnet	Bot	143097	0.757	100168	42929	6.335%
Brute Force	FTP	25933	0.137	18153	7780	1.148%
	SSH	94979	0.503	66485	28494	4.205%
Infilteration	Infilteration	116361	0.616	81453	34908	5.152%
Dos	Hulk	432648	2.290	302854	129794	19.154%
	SlowHTTPTest	14116	0.075	9881	4235	0.625%
	GoldenEye	27723	0.147	19406	8317	1.227%
	Slowloris	9512	0.050	6658	2854	0.421%
Web Attacks	Web	2143	0.011	1500	643	0.095%
	XSS	927	0.005	649	278	0.041%
	SQL Injection	432	0.002	302	130	0.019%
Total		18893270	100	1581141	677632	100%

Evaluation metrics

Various measurement metrics and tools are used to evaluate the results in problems where ML models are used. In this way, it is possible to evaluate the performance and deficiencies of the IDS systems. Explanations of the metrics are given in this section.

Table 2. Confusion matrix of binary classification

		Predicted Class	
		Attack	Benign
Actual Class	Attack	TP	FN
	Benign	FP	TN

TP = True positives, TN = True Negatives, FP = False positives, FN = False Negatives

The confusion matrix is a classification table based on actual class (exact reference) in the rows and the class projected on it in the columns as shown in Table-2. This table can be produced in both multiclass and binary classification problems. It shows the classification distribution of a model and helps determine the situation in which one class is constantly classified as incorrectly in another class. For example, a DDos attack can be classified as a continuous exfiltration. The researchers can then examine the possible causes of the problem. These situations regarding models in many different metrics cannot be easily observed.

Using this table, important metrics such as acc, precision, recall and f1-measure are calculated. Thus, it is possible to evaluate the classification performance from different angles.

For multi class problems, these metrics should be calculated separately on a class basis. If it desired to obtain a general value regarding the classification performance, it can be done by using following methods.

• Micro average: Calculates the final value using the sum of the classification metrics for each class (TP = TPClass1 + TPClass2 + …)

• Macro average: Metrics per class are calculated and averaged. It refers to the process of adding them all and dividing them by the sum of the number of classes.

• Weighted macro average: It is a similar method to calculate macro average, but when obtaining the overall metric value of the model, the number of samples in the classes is also taken into account in the average calculation. All classes are not included in the overall average with equal weight.

Multi class classification

IDS 2018 dataset contains a total of 15 classes, 14 of which are malicious and 1 benign. The first stage of the tests of the model proposed in this study was conducted for the classification of these data. The test results are given in Table-3 in detail.

Table 3. Multiclass classification rate (before and after proposed pre-processing model)

MULTICLASS (15 CLASS) CLASSIFICATION RATE COMPARISON								
	Before Proposed Pre-processing				After Proposed Pre-processing			
	Acc	Prec.	Recall	F score	Acc	Prec.	Recall	F score
Logistic Reg (LR)	0.949	0.937	0.949	0.940	0.989	0.989	0.989	0.989
RF	0.998	0.998	0.998	0.998	0.990	0.990	0.990	0.989
DT	0.998	0.998	0.998	0.998	0.990	0.990	0.990	0.989
LDA	0.965	0.972	0.965	0.967	0.959	0.981	0.959	0.968
ExtraTree	0.998	0.998	0.998	0.998	0.990	0.990	0.990	0.989
XGB	0.996	0.996	0.995	0.997	0.955	0.987	0.955	0.965
GB	0.997	0.997	0.997	0.997	0.955	0.987	0.955	0.965
SVC	0.950	0.947	0.952	0.940	0.990	0.990	0.990	0.989
KNN	0.949	0.937	0.949	0.940	0.989	0.989	0.990	0.990

With the datasets of 15 classes, the tests were tested with 9 different machine learning techniques. When the tests performed with original dataset, a very successful classification performance of 99%. However, high classification rates were obtained in similar studies on this subject. Therefore, the purpose of conducting these tests is to observe the effect of the proposed model on the classification performance. When the results were evaluated, although there was a loss of data during proposed pre-processing, the results were found to be better for some classifiers and similar values for some others.

The required processing times in the experiments are shown separately for each process step in Tablo-4.

Table 4. Multiclass classification time comparison (before and after pre-processing)

MULTI CLASS (15 CLASS) TIME COMPARISON (seconds)		
Extra Tree Algorithm	Before Proposed Pre-processing (original data) (sec.)	After Proposed Pre-processing (sec.)
Loading Data	699.6770	63.0300
Label Encoding	5.4992	0.0035
Under Sampling	743.1130	148.8480
Data Split for Training and Test	9.4730	0.7700
Training time	422.7824	330.1324
Testing time	27.2800	16.5900
Testing time for each sample	0.00004	0.00002

In Table-4, separate time periods are calculated for the process steps that may be needed in a sample machine learning model design. A much faster process is carried out thanks to the proposed pre-processing model in all steps such as reading data, label encoding, under sampling, and classifying data. In addition, more importantly, an average of 64.43% shortening is achieved in the test time per sample. Considering that IDS systems monitor internet traffic in real time and there is an average of 30 thousand or 50 thousand packages per second, the test time of each sample is very important. Thanks to the model proposed in this study, the test time per sample is considerably shortened, and this makes it more possible to use in real-time analysis tools. It this study, it is aimed to make the model with very high classification rate with a shorter processing time. The results support this goal.

Binary classification

In some IDS architectures, traffic differs only as malicious or benign. Malicious traffic is not expected to be categorized separately. In order to simulate this problem, the tests were repeated as binary classification. The obtained results are shown in Table-5.

Table 5. Binay classification rate (before and after proposed pre-processing model)

BINARY (BENIGN- MALICIOUS) CLASSIFICATION RATE COMPARISON								
	Before Proposed Pre-processing (original data)				After Proposed Pre-processing (original data)			
	Acc	Prec.	Recall	F-score	Acc	Prec.	Recall	F score
Logistic Reg (LR)	0.877	0.917	0.829	0.871	0.963	0.983	0.94	0.962
RF	0.975	0.978	0.971	0.975	0.975	0.996	0.95	0.974
DT	0.971	0.970	0.972	0.971	0.975	0.996	0.95	0.974
GNB	0.895	0.965	0.820	0.886	0.905	0.946	0.86	0.901
LDA	0.942	0.960	0.922	0.940	0.946	0.968	0.92	0.944
AdaB	0.976	0.994	0.958	0.976	0.957	0.973	0.94	0.956
ExtraTree	0.974	0.977	0.971	0.974	0.985	0.996	0.96	0.984
GB	0.980	0.999	0.961	0.980	0.968	0.992	0.94	0.967
XGB	0.975	0.991	0.948	0.977	0.975	0.996	0.95	0.974

In this study, a new pre-processing method is proposed to shorten the processing time for classification. It is expected that the shortening of the processing time will not reduce the classification performance. For this purpose, the tests were repeated in the same environments. When the test results related to binary classification are examined, it is seen that generally successful results are obtained as in multi class classification. It has been observed that the pre-processing does not cause any loss in performance. The advantage provided in terms of processing time in the classification shown graphically in Figure-4.

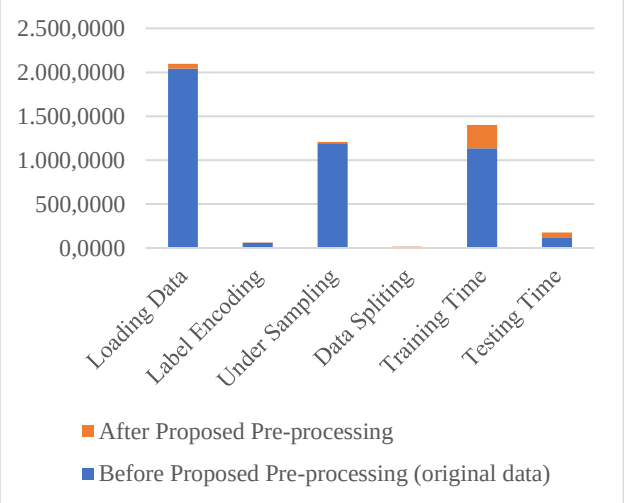

Figure 4. Binary classification time comparison (before and after pre-processing).

If a classification process is divided into steps as shown in Figure-4, the processing times were compared for each processing step. As a result, it is expected that the proposed model needs shorter processing times in all steps. Accordingly, as can be seen in the graph, after the proposed models, considerably shorter processing times were needed than traditional models.

In general, the results show that the direct equivalents of the values obtained from the flow of traffic in IDS systems do not have any meaning in the classification models, there are sharp distinctions between the classes in terms of traffic

characteristics, therefore, it is understood that using the value in a simplified form makes a significant contribution to the processing time.

Comparison with similar works and discussion

The classification results obtained in the studies with the same dataset used in this work are shown comparatively in Table-6. In the test results for multiclass classification were at the same level with other studies. Classification success was achieved with 99%.

In complex and heavy network traffic flows, it is very important to be fast as well as accuracy in classification of it. In the tests performed with proposed model, similar classification rated were achieved with other studies using mush shorter processing time. Thus, the classification was possible both quickly and with high accuracy.

Table 6. Comparison with similar works using CSE CIC IDS 2018 dataset

Paper - Year	Dataset	Model Details	Results
Yu et al. [27] - 2021	CIC-IDS 2018, CIC-IDS 2017	Hierarchical packet byte-based CNN (PBCNN)	Acc: 0.999 Precision: 0.982 Recall: 0.983 F1-score: 0.983
Vinayakumar et al. [28]-2019	CIC-IDS 2018, CIC-IDS 2017	ANN 5 hidden layer (1024, 768, 512, 256, 128 node), Relu Activation Function	Acc: 0.962 Precision: 0.962 Recall: 0.965 F1-score: 0.957
Ferrag et al. [29]-2020	CIC-IDS 2018	DNN, RBM, DBFN, CNN, DBM, Deep autoencoder	Acc: 0.9728
Proposed model with original dataset	CIC-IDS 2018	RF, DT, ExtraTree	Acc: 0.998 Precision: 0.998 Recall: 0.998
Proposed model with pre-processed data	CIC-IDS 2018	RF, DT, ExtraTree	Acc: 0.990 Precision: 0.990 Recall: 0.990 F1-score: 0.989

Conclusion and Future Work

In this paper, detailed experimental results of CSE-CIC-IDS2018, which is a modern dataset for the application of ML to network intrusion systems, are shown. Model design and application details are explained. Tests were carried out with 9 different machine learning algorithms in the type of supervised. Necessary optimization processes were applied in the parameter selection of algorithms. In general, it was determined that tree-based classifiers performed best. Classifiers based on decision trees achieved a certain success value in all classes. For this reason, they have become practically applicable. The most increases were provided for the ExtraTree and SVC classifiers.

In todays, internet traffic is increasing and diversifying due to the increase in internet usage. For this reason, there is a need for tools/analyzers that perform much faster and accurate traffic classification and this need is increasing day by day. With the proposed model in this study, it has been shown that complex traffic can be made much simpler and used in classification. In the future studies, performing analyzes using fewer features will be beneficial in terms of speed. In addition, using artificial neural network structures will allow successful results. It is possible to get more accurate results with more complex learning structures. Finally, unsupervised learning methods can be used that can adapt itself to new attack types (zero day attacks). So that it will always be possible to create up-to-date models.

Conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared. There is no conflict of interest with any person / institution in the article prepared.

Acknowledgement

We special thank to Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani for providing CSE-CIC-IDS2018 Dataset publicly.

References

[1] McKinney Wes, “Data structures for statistical computing in Python”, Proceedings of the 9th python in science conference, 1-6, 2010.

[2] Pedregosa F, Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O, et al., “Scikit-learn: machine learning in Python”, Journal of Machine Learning Research 12, 2825-2830, 2011.

[3] Chen T., Guestrin C., “Xgboost: a scalable tree boosting system”, Proceedings of the 22nd ACM SIGKDD International conference on Knowledge Discovery and Data Mining, 785-794, August, 2016.

[4] CyberEdge, 2021. 2021 Cyberthreat Defense Report. https://cyber-edge.com/cdr/

[5] FireEye, 2021. M-trends 2021Cyber Security Report. FireEye, https://www.fireeye.com/blog/threat-research/2021/04/m-trends-2021-a-view-from-the-front-lines.html

[6] Liao H-J, Richard Lin C-H, Lin Y-C, Tung K. “Intrusion detection system: A comprehensive review”, Journal of Network and Computer Applications, 36(1), 16-24, 2013.

[7] Sunanda Gamage, Jagath Samarabandu, "Deep learning methods in network intrusion detection: a survey and an objective comparison", Journal of Network and Computer Applications, 169, 1-21, 2020.

[8] Ansam Khraisat, Ammar Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges", Cybersecurity, 4(18), 1-27, 2021.

[9] C Kalimuthan, J. Arokia Renjit, "Review on instrusion detection using feature selection with machine learning techniques", Materials Todat: Proceddings, 33(7), 3794-3802, 2020.

[10] Cisco, 2021. Cisco Security Analytics Whitepaper. https://www.cisco.com/c/dam/en/us/products/collateral/security/stealthwatch/sw-siem-optimization-wp.pdf.

[11] Darktrace, 2021, Preparing for AI-enabled Cyberattacks, Whitepaper, https://www.darktrace.com/en/mit-preparing-for-cyberattacks/.

[12] Vectra Cognito, 2020. How to Augment Security Operations Center with Artificial Intellignece, Whitepaper, https://content.vectra.ai/rs/748-MCE-447/images/WhitePaper_AugmentSOCwithAI.pdf.

[13] Chencheng MA, XueHui Du, Lifeng Cao, "Analysis of Multi-Types of Flow Features Based on Hybrid Neural Network for Improving Network Anomaly Detection", IEEE Access, 7, 1-18, 2019.

[14] Lan Liu, PengCheng Wang, Jun Lin, LangZhou Liu, "Intrusion Detection of Imbalanced Network Traffic based on Machine Learning and Deep Learning", IEEE Access, 9, 1-14, 2021.

[15] University of New Brunswick (UNB). A realistic cyber defense dataset (CSE-CIC-IDS2018), https://www.unb.ca/cic/datasets/ids-2018.html.

[16] Iman Sharafaldin, Arash Habibi Lashkari, Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy, 1-9, 2018.

[17] G. Karatas , O. Demir , O.K. Sahingoz , Increasing the performance of machine learn- ing-based IDSs on an imbalanced and up-to-date dataset, IEEE Access 8 (2020) 32150–32162 .

[18] Hongpro Zhang, Lulu Huang, Chase Q. Wu, Zhanbo Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset", Computer Networks, 177, 1-10, 2020.

[19] V. Kanimozhi, T. Prem Jabob, "Artificial Intelligence based Network Intrusion Detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing", ICT Express, 5, 211-214, 2019.

[20] Yesi Novaria Kunang, Siti Nurmaini, Deris Stiawan, Bhakti Yudho Suprapto, "Attack clasification of an instrusion detection system using deep learning and hyperparameter optimization", Journal of Information Security and Applications, 58, 1-15, 2021.

[21] S. S. Volkov, I I Kurochkin, "Network attacks classification using Long Short-term memory based neural networks in Software Defined Networks", 9th International Young Scientist Conference on Computational Science, 178, 394-403, 2020.

[22] Mahendra Prasad, Sachin Tripathi, Keshav Dahal, "Unsupervised feature selection and cluster center initialization based arbitrary shaped clusters for intrusion detection", Computers & Security, 99, 1-19, 2020.

[23] Jiyeon Kim, Jiwon Kim, Hyunjung Kim, Minsun Shim, Eunjung Choi, "CNN-based network intrusion detection against Denial-of-Service Attacks", Mdpi electronics, 1-21, 2020.

[24] Mokhtar Mohammadi, Tarik A. Rashid, Sarkhel H.Taher Karim, Adil Hussain Mohammed Aldalwie, Quan Thanh Tho, Moazam Bidaki, Amir Masoud Rahmani, Mehdi Hosseinzadeh, "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems", Journal of Network and Computer Applications,178, 1-23, 2021.

[25] T. Daniya, K. Suresh Kumar, B. Santhosh Kumar, Chandra Sekhar Kolli, "A survey on anomaly-based intrusion detection system", Materials Today: Proceedings, 1-4, 2021.

[26] Arwa Aldweesh, Abdelouahid Derhab, Ahmed Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues", Knowledge-Based Systems, 189,1-19, 2020.

[27] Lian Yu, Jingtao Dong, Lihao Chen, Mengyuan Li, Bingfeng Xu, Zhao Li, Lin Qiao, Lijun Liu, Bei Zhao, Chen Zhang, "PBCNN: Packet Bytes-based Convolutional Neural Network for Network Intrusion Detection", Computer Networks, 1-24, 2021.

[28] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," in IEEE Access, vol. 7, pp. 41525-41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[29] Mohamed Amine Ferrag, Leandros Maglaras, Sotiris Moschoyiannis, Helge Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study", Journal of Information Security and Applications, 50, 1-19, 2020.

[30] Mesut Uğurlu, İbrahim Alper Doğru, Recep Sinan ARSLAN, "A new classification method for encrypted internet traffic using machine learning", Turkish Journal of Electrical Engineering and Computer Sciences, Accepted. 2021