ISSN: 0067-2904

Intrusion Detection System Using Data Stream Classification

Amer Abdulmajeed Abdulrahman*[1], Mahmood Khalel Ibrahem[2]
[1]Informatics Institute for Post Graduate Studies, College of Science, University of Baghdad, Baghdad, Iraq
[2]College of Information Engineering, Al-Nahrain University

Received: 8/12/2019 Accepted: 15/3/2020

Abstract

Secure data communication across networks is always threatened with intrusion and abuse. Network Intrusion Detection System (IDS) is a valuable tool for in-depth defense of computer networks. Most research and applications in the field of intrusion detection systems was built based on analysing the several datasets that contain the attacks types using the classification of batch learning machine. The present study presents the intrusion detection system based on Data Stream Classification. Several data stream algorithms were applied on CICIDS2017 datasets which contain several new types of attacks. The results were evaluated to choose the best algorithm that satisfies high accuracy and low computation time.

Keywords: Intrusion Detection System, Data Stream Classification, CICIDS 2017 dataset, Feature selection

نظام كشف الاختراق باستخدام تصنيف دفق البيانات

عامر عبد المجيد*، محمود خليل
[1]معهد المعلوماتية للدراسات العليا ، كلية العلوم ، جامعة بغداد ، بغداد ، العراق
[2]كلية هندسة المعلومات، جامعة النهرين، بغداد، العراق

الخلاصة

يتم دائمًا تهديد اتصالات البيانات الآمنة عبر الشبكات بالتسلل وإساءة الاستخدام. يعد نظام كشف التسلل عن الشبكات أداة قيمة للدفاع المتعمق لشبكات الكمبيوتر. معظم الأبحاث والتطبيقات في مجال نظام منع وكشف الاختراق تستند على أساس تحليل مجموعات البيانات العديدة التي تحتوي على أنواع الهجمات باستخدام تصنيف التعلم الدفعي. قدم هذا البحث نظام كشف الاختراق استنادًا إلى تصنيف تدفق البيانات. تم تطبيق العديد من خوارزميات تدفق البيانات على مجموعات بيانات CICIDS2017 التي تحتوي على عدة أنواع جديدة من الهجمات. تم تقييم النتائج لاختيار أفضل خوارزمية تحقق الدقة العالية واقل وقت حساب

1. Introduction

Intrusion detection system (IDS) has played a pivotal role in defending the networks by directing security officials to warn them about malignant behaviors such as attacks, malware, and intrusions. The presence of IDS is a compulsory line of defense to protect vital networks from these ever-increasing issues of intrusive activities. Therefore, research in the field of IDS has flourished over the years to suggest better IDS systems. However, many researchers are struggling to find valid and comprehensive datasets that enable testing and evaluating their proposals; the major challenge in itself is having an appropriate dataset [1].

*Email: amer6567@yahoo.com

Recently, the data stream model has appeared to resolve the continuous data issue. Naturally, written data streams algorithms can handle sizes of data much larger than memory, and can be extended to challenging real time applications that have not been handled through data mining or machine learning.

The basic data stream processing assumption involves the examination of the training examples only for a short time. That is, they reach in a high speed stream and should be ignored to make room for next examples. The data stream algorithm does not have control over the order of the examples shown, and its model should be updated gradually when each example is examined. Property at any time is required that the model be ready for application at any time between training examples [2].

Classification algorithms of data stream require complete and appropriate evaluation practices. The assessment must allow users to ensure that certain issues can be addressed, identify improvements to algorithms, and determine which algorithms are most appropriate to their problem [3].

Measuring the performance of data stream classification is a two dimensional problem involving accuracy and processing speed (time).

In this paper, the contributions are twofold. Firstly, analyzes of the CICIDS2017 datasets to reduce high class imbalance problem and preprocess these datasets. Secondly, selection of the important feature sets to detect different attacks and implement several common data stream machine learning algorithms to evaluate the algorithm that selects the distinguished features.

2. Related works

Akanksha *et al.* (2017) used different streaming data mining classification techniques to improve the efficiency of the IDS. They applied and compared their results based on Naïve Bayes, Hoeffding tree, Accuracy Updated Ensemble and Accuracy Weighted Ensemble data stream classification algorithms on NSL-KDD datasets. The results showed that the best classifier was the Naive Bayes, with higher accuracy but longer time, whereas Hoeffding tree classifier showed accuracy nearest to that of the Naive Bayes classifier but with shorter time [4].

Loo Hui Ru *et al*. (2014) proposed an algorithm for classifying online data streams and learning with limited labels using selective semi-supervised training classification. They used KDD'99 and Cambridge datasets to create the model. The cumulated accuracy for the proposed classification method is up to 97% and 99% for Cambridge and KDD'99 datasets, respectively[5].

Czarnowski and Piotr (2014) proposed and validated a new approach to mine data streams with concept drift using the ensemble classifier created from the single class base classifiers. It is assumed that base classifiers of the proposed ensemble are induced from incoming portions of the data stream. Several datasets were compared for best evaluation [6].

Saddam and Anirudh (2018) explored the performance of network intrusion detection system (NIDS) which can detect various types of attacks in the network using Deep Reinforcement Learning Algorithm. They exploited Deep Q Network algorithm which is a value-based Reinforcement Learning algorithm technique used in the detection of network intrusions. Moreover, they analyzed the accuracy of their model in comparison with different types of attacks. In their paper, they illustrated the comparison of their NIDSDQN model to a previous model designed in other approaches such as J48, artificial neural network, random forest, and support vector machine. They worked on CICID2107 datasets which aided as an effective means in the detection of different types of attacks. The results of Deep Q Network-Intrusion Detection System model demonstrated improvement in the accuracy and performance. The accuracy values were 95.53%, 92.32% and 89.245% using DDoS, Port Scan and Infiltration attacks, respectively [7].

3. CICIDS2017 datasets

CICIDS2017 datasets are generated by the Canadian Institute for Cybersecurity. Each dataset contains benign and the most up-to-date common attacks such as DoS, DDoS, brute force SSH, brute force FTP, heartbleed, infiltration, and botnet, which make it the most up-to-date as compared to other datasets. These datasets also include analyzing network traffic results based on IP source and IP destination, source and port destination port, time stamp, protocols, and attacks [8].

CICIDS2017 datasets are designed for intrusion detection and network security purposes. There are several attack profiles created based on the latest updated list of common attack families and implemented with related tools and codes.

The main types of attack profiles are:

- **Distributed Denial of Service DDoS Attack:** This usually occurs over victim resources, multiple systems, or bandwidth overwhelms. This attack is often the result of multiple hacked systems (e.g. botnet) flooding the target system by generating the massive network traffic [9].
- **Port Scan attack:** this attack sends client requests to a set of server port addresses on a host, intended to find an active port and exploit known security sensitivity for that service. Surveying, as a way to discover exploitable communication channels, has existed throughout the ages. The idea is to investigate as many listeners as possible and track down recipients or beneficiaries for user's own need [10].
- **Botnet:** Number of internet connected devices used by the owner of robots to perform different tasks. It can be used to send spam, steal data, and allow an attacker to access and connect to the device [11].
- **Web Attack**: These types of attacks come out on daily basis, because individuals and organizations are currently taking serious measures of security. We use the SQL Injection, by which an attacker can create a series of SQL commands, for forcing the database to respond to the information We also employ Cross-Site Scripting (XSS), that occurs when developers do not properly test their code to find the ability to inject script, and Brute Force over HTTP which can try the list of passwords to find the administrator password [8].
- **Infiltration Attack:** Internal network infiltration often exploits vulnerable software such as Adobe Acrobat Reader. After successful exploitation, the tailgate will be executed on the victim's computer and can perform various attacks on the victim's network such as full port scanning, IP sweep, and number service, using Nmap [8].

Unlike other IDS datasets that separate training from testing data, CICIDS2017 gathers all labeled records of each specified type of attacks into a unique CSV file format. Each CSV file is composed of a given number of labeled records, along with 85 features that describe these records. Table-1 shows the 85 features and their data type (numerical or nominal).

Table 1-Feature names and data types of CICIDS21017 dataset

Feature name	Data type
Flow ID, Source IP, Destination IP, Timestamp	Nominal
Source Port, Destination Port, Protocol, Flow Duration, Total Fwd Packets, Total Backward Packets,Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length Max, Fwd Packet Length Min, Fwd Packet Length Mean, Fwd Packet Length Std,Bwd Packet Length Max, Bwd Packet Length Min, Bwd Packet Length Mean, Bwd Packet Length Std,Flow Bytes/s, Flow Packets/s, Flow IAT Mean, Flow IAT Std, Flow IAT Max, Flow IAT Min,Fwd IAT Total, Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, Fwd IAT Min,Bwd IAT Total, Bwd IAT Mean, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min,Fwd PSH Flags, Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, Fwd Header Length1, Bwd Header Length,Fwd Packets/s, Bwd Packets/s, Min Packet Length, Max Packet Length, Packet Length Mean, Packet Length Std, Packet Length Variance,FIN Flag Count, SYN Flag Count, RST Flag Count, PSH Flag Count, ACK Flag Count, URG Flag Count, CWE Flag Count, ECE Flag Count, Down/Up Ratio, Average Packet Size, Avg Fwd Segment Size, Avg Bwd Segment Size, Fwd Header Length,Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk,Bwd Avg Bulk Rate,Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets, Subflow Bwd Bytes, Init_Win_bytes_forward, Init_Win_bytes_backward, act_data_pkt_fwd, min_seg_size_forward, Active Mean, Active Std, Active Max, Active Min,Idle Mean, Idle Std, Idle Max, Idle Min	Numerical

It can be seen from the web attacks dataset (Table-2) that the majority class of prevalence was the class Benign (98.72 %) while the values for the minority classes Brute force , XSS and Sql injection were (0.88%), (0.38%) and (0.01%) respectively. In such a large difference of propagation rate, the potential detector may be leaning towards benign. This situation becomes a high-grade imbalance when the dataset is used for the training of classification or detection. There are many ways to address the problem of class imbalance for a dataset [12]. One of the major approaches is to rename classes

that include splitting majority classes to compose more classes or merging few minority classes to compose a class, thus improving the prevalence ratio and reducing class imbalance issue. We are relabeling all types of web attack classes to one class.

Table 2-Number and percentage labels for each CSV file.

Dataset	No of instance	Benign	No of attack	
DDoS attack	225,745	183,910 83.47%	41,835 18.53%	
Port-scan	286467	127,537 44.5%	158,930 55.5%	
Botnet	191,033	189,067 99%	1,966 1%	
Infiltration	288566	288527 99,99%	39 0.01%	
Web Attacks	170366	168,186 98.72%	Brute force	1507 (0.88%)
			XSS	652 (0.38%)
			Sql injection	21 (0.01%)

Table-3 illustrates the CICIDS2017 features distribution of each attack based on protocol types. This table shows that the TCP protocol series is the most launched attack by attackers. The TCP protocol is easy and clear to be used by attackers to place network-based attacks on victim computers.

Table 3-CICIDS2017 features distribution based on protocol types

	BENIGN	DDoS	BENIGN	botnet	BENIGN	Infiltration	BENIGN	web	BENIGN	Port-scan
UDP	54	0	152	0	285	0	141	0	95	6
TCP	150985	41835	93233	1966	186133	36	86275	2180	67334	158923
HTTP	32871	0	95682	0	102148	0	81770	0	60108	1

4. Data stream learning techniques

Data stream is an unlimited and ordered sequence of instances that arrive over time. It puts specific limitations on the learning system that cannot be met by legal algorithms from this field. Learning data stream assumes the following: [2]

- Arriving data are one by one.
- Data point's number is unlimited.
- Data distribution changes over time.
- Training and Testing are overlapping. The machine learning system can train from the previous test points.
- In general, data processing rate must be higher than data arrival rate.
- The learning algorithm requirements and the space used must be firmly linked.

In data mining, data stream classification is a special type to classify data streams. The major requirement for performing this classification is the ability to simultaneously learn and classify the arriving data. Traditional data mining classifiers, such as decision tree based on batch training, require a large batch of data before performing the training. When concept drift occurs, retraining is needed [13].

The main methods to tackling data streams classification are:

• **Sliding windows.** The assumption is maintaining a fixed-size buffer to the latest examples. These windows are used for classification and then discarded as new instances become available. This allows tracking the proceeding of the data stream by storing the current state in memory. This is achieved by either removing the oldest cases or weighting them dynamically according to their suitability. The window size has a decisive effect on its performance. A large window can save more information efficiently, but it may include instances of different concepts. While a small window is able to adapt for fast and small changes, it may lose the overall context of the analyzed problem and is susceptible

to overloading. Recent studies solve this problem by using multiple windows at the same time or focusing on adjusting size dynamically. A properly defined sliding window will be able to adapt to changes in the data stream. This is known as implied drift handling [2].

• **Online learners** update on instance by instance basis, which leads to absorbing the changes in flow as they occur. The requirements of these models must be met as follows,

- During training, each instance should be processed only once.
- Computational complexity of handling each instance must be as small as possible.
- Accuracy should not be less than that of a classifier trained on batch data collected up to the certain time.

In the online mode, some of standard classification algorithms may operate, e.g. Naïve Bayes or Artificial Neural Networks. However, there is a large number of modified methods to provide effective online mode. These methods also provide implied drift handling [2].

• Ensemble learners: An ensemble can be described as an aggregation of many weak learners to form one strong learner that has a high prediction of performance. Random Forests, Boosting, and Bagging are samples of ensemble methods that achieve higher learning performance. The Random Forests learner train decision trees on resampled versions of the original data, then randomly selecting a small number of features that can be examined at each node for split. The Boosting learner trains classifiers iteratively with increasing the weight of instances that were previously misclassified. The Bagging learner uses resampling to train classifiers on different subsets of instances, which effectively increases the variance of each classifier without increasing the overall bias [14].

There are multiple versions of Boosting and Bagging that are part of the current methods for evolving data stream learning, such as OZABoosting and Leveraging Bagging. Adaptive Random Forests algorithm (ARF) is considered as a new streaming classifier for evolving data streams[15].

• **Stochastic Gradient Descent (SGD)** is popular and has proven to have high performance in a variety of machine learning tasks. It includes a learning rate parameter to determine the length of the next step to take when moving forward in the gradient direction [16].

5. Data preprocessing

Data preprocessing is the major phase within the knowledge discovery process. Pre-processing data involve more time and effort in the complete data analysis. Usually, metadata come with many disadvantages such as missing values, noise, inconsistencies, and redundancies. Thus, low quality data cause the performance of subsequent learning algorithms to be undermined. The suitable preprocessing steps can be able to significantly influence the reliability and quality of next automatic decisions and discoveries. Data preparation is a part of pre-processing that aims to convert the primary input into high quality one, which is suitable for the mining process to be followed. Preparation is a compulsory step that includes techniques such as normalization, transformation, cleaning, and integration [17].

Most of the available datasets contain unwanted elements (missing, redundant, or infinite values) that should be removed or transformed. The step of preprocessing is essential to obtain a suitable dataset.

6. Data reduction

In data mining, data reduction is an important step before processing that allows for a fast, adaptable, and accurate model that is characterized by low complexity of computation with fast response to changes and incoming objects. Reducing the incoming data complexity dynamically is critical to evolve these models. In addition, due to the existence of the concept of drifting, the number and importance of features and instances may change over time. We should also consider this while updating and maintaining an online model.

Data reduction techniques in data stream learning scenarios are required to process items over the online mode as quickly as possible without making any assumptions about prior distribution of data [18].

One of dimension reduction technique types is feature selection. It has been proven to be efficient and effective in dealing with high dimensional data. It directly selects a subset of features related to model construction. Since feature selection retains a subset of the original features, one of its main properties is that it well preserves the physical meanings of the original feature sets and provides better readability and interpretability. Due to this particular reason, it is more widely applied in many real world applications such as text extraction and gene analysis. Feature selection obtains relevant features

by removing redundant and unnecessary features. Removing these redundant and unnecessary features reduces the storage and computation costs without significant loss of information or negative deterioration of the learning performance [19].

The categories of feature selection methods can be classified into: [19]

- Filtering methods: The selection is based on data related measures, such as crowding or reparability.
- Embedded methods: The optimal features subset is built in the classifier construction.
- Wrapper methods: The selection criterion is part of the fitness function and therefore depends on the learning algorithm.

7. Experimental results

Similar to all datasets, CICIDS2017 datasets contain unwanted elements (missing, redundant or infinite values) that should be removed or transformed. It was necessary to clean up these datasets from errors which could occur while flow data are being acquired. The following steps include the cleanup preprocessing and implementing work:

- First step: redundant records were dropped from the whole dataset. All missing values were replaced by zeros and infinite values were replaced by the mean of their attribute value. CICIDS2017 datasets contain features that were recorded while acquiring data flow. Those features are related to a specific network and do not have any impact on model results.
- Second step of the dataset preprocessing consists of removing all those meaningless features manually in order to decrease the data dimension. Among the removed useless features related to specific network are Flow ID, Source IP, Destination IP, Source Port, Destination Port and Timestamp. By removing them, nominal features processing disappears since some classification models require numerical values rather than nominal ones.
- Third step of preprocessing is removing features with low standard deviation. In this experiment, standard deviation criterion is used to remove all features with standard deviation value equal to zero, since removing those increases the model's accuracy. Also, those features are irrelevant in data and can decrease the performance of the model analysis. By applying the standard deviation removing criteria, ten features were eliminated from datasets, which are Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, CWE Flag Count, Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk and Bwd Avg Bulk Rate. The number of attributes from the above steps is show in Table- 4 as selected#1.
- Fourth step included the selection of the highly important features for each dataset by using C5.0 algorithm. This algorithm is considered one of the embedded methods for feature selection which gives the optimal features subset. The number of attributes in this step is shown in Table-4 as selected#2. Table-4 shows the abstraction number of features after data preprocessing and data reduction.

Table 4-Summary of features after data preprocessing and data reduction

	DDoS attack	Port-scan	Botnet	Infilteration	Web Attacks
Original	*85*	*85*	*85*	*85*	*85*
Selected #1	*69*	*69*	*69*	*69*	*69*
Selected #2	*26*	*13*	*16*	*8*	*10*

- Fifth step describes the preparation of the simulation and the results of the work. The experiment is conducted to explore the ability of the data stream classifiers to learn accurately. Real concept drifts datasets, DDoS attack, Port-scan, Botnet, Infiltration, and Web Attacks are chosen for the experiment. In this experiment, the first 1000 instances were used in the training stage, then the rest of the data were labeled randomly. The interleave test-then-train method was used to verified the accuracy of each model, where the data were first tested before being gradually trained. The mean time and cumulated accuracy results of the evaluation metrics were computed for the five selected common data stream machine learning algorithms, namely Naive-Bayes (NB), Adaptive Random Forest (ARF), Multilayer Perceptron (MLP), OzaBoost, and Stochastic Gradient Descent (SGD).

Tables- 5 and 6 show the results of cumulated accuracy and mean time for 69-feature selection and impotent C5.0 feature selection, respectively, for each data stream algorithm.

Figures- 1, 2, 3, 4 and 5 illustrate the cumulative accuracy result chart, which represents the percentage of the total correct prediction on each segment.

Table 5-Cumulated accuracy and mean time with 69-feature selection

	NB		**ARF**		**MLP**		**OzaBoost**		**SGD**	
	Acc	***T/sec***	***Acc***	***T/sec***	***Acc***	***T/sec***	***Acc***	***T/sec***	***Acc***	***T/sec***
DDoS attack	*60.66*	*3.51*	*99.90*	*14.52*	*87.43*	*2.87*	*97.71*	*58.8*	*99.73*	*2.77*
Port-scan	*72.52*	*3.81*	*99.97*	*30.24*	*67.71*	*3.62*	*99.92*	*51.17*	*97.27*	*2.15*
Botnet	*67.40*	*2.71*	*99.83*	*17.70*	*96.66*	*2.48*	*99.87*	*44.75*	*99,11*	*2.38*
Infiltration	*83.60*	*4.17*	*99.99*	*18.86*	*99.89*	*3.54*	*99.96*	*38.68*	*99.92*	*3.46*
Web Attacks	*50.21*	*2.47*	*100*	*9.50*	*92.69*	*2.07*	*100*	*44.75*	*99.99*	*2.16*

Table 6-Cumulated accuracy and mean time with C5.0 feature selection

	NB		**ARF**		**MLP**		**OzaBoost**		**SGD**	
	Acc	***T/sec***	***Acc***	***T/sec***	***Acc***	***T/sec***	***Acc***	***T/sec***	***Acc***	***T/sec***
DDoS attack	*62.21*	*1.45*	*99.91*	*9.73*	*89.61*	*1.32*	*98.01*	*10.06*	*99.72*	*1,08*
Port-scan	*46.38*	*0.84*	*99.97*	*10.32*	*68.66*	*0.84*	*99.97*	*9.07*	*98.66*	*0.64*
Botnet	*71.01*	*0.72*	*99.90*	*8.41*	*94.50*	*0.69*	*99.95*	*8.37*	*98.90*	*0.59*
Infiltration	*85.45*	*0.65*	*99.99*	*7.72*	*99.95*	*0.59*	*99.98*	*5.79*	*99.94*	*0.58*
Web Attacks	*58.42*	*0.43*	*100*	*4.69*	*99.94*	*0.46*	*100*	*2.24*	*100*	*0.34*

In Table-5, we can see the different results of cumulated accuracy. Overall, SGD, OZAboost, and ARF had similarly high values for all datasets, although ARF performed slightly better. But, interestingly, the results took a long time to implement, which is not appropriate in any system to detect intrusion. Table-6 shows the reduction of time when the C5.0 feature selection was used with saving the same accuracy approximately.

Tables-5 and 6 show that the best model for all datasets used is SGD, because it has performed in less time with high accuracy.

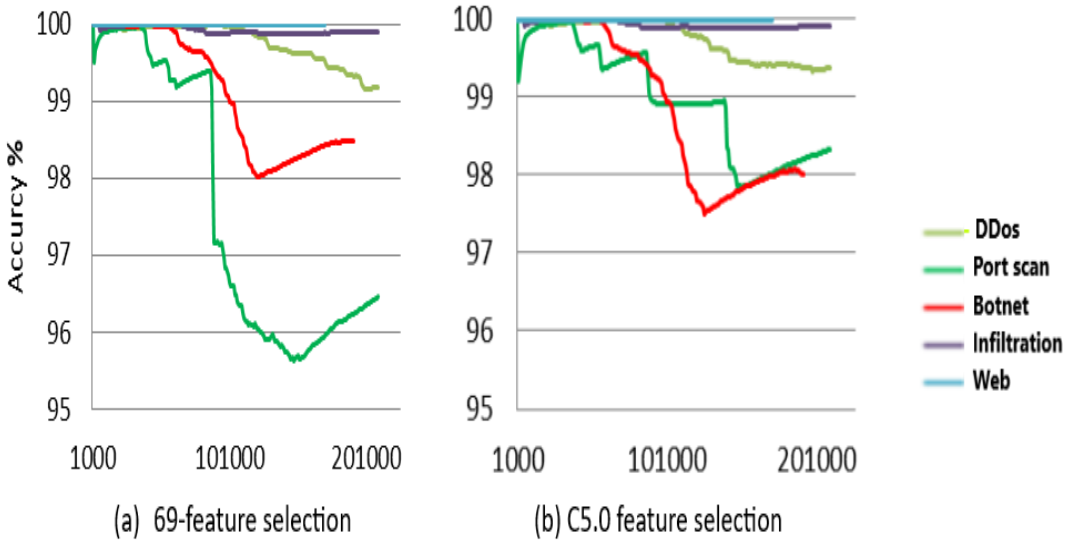

Figure 1-Comparison of cumulated accuracy in SGD with all CICIDS2017 datasets.

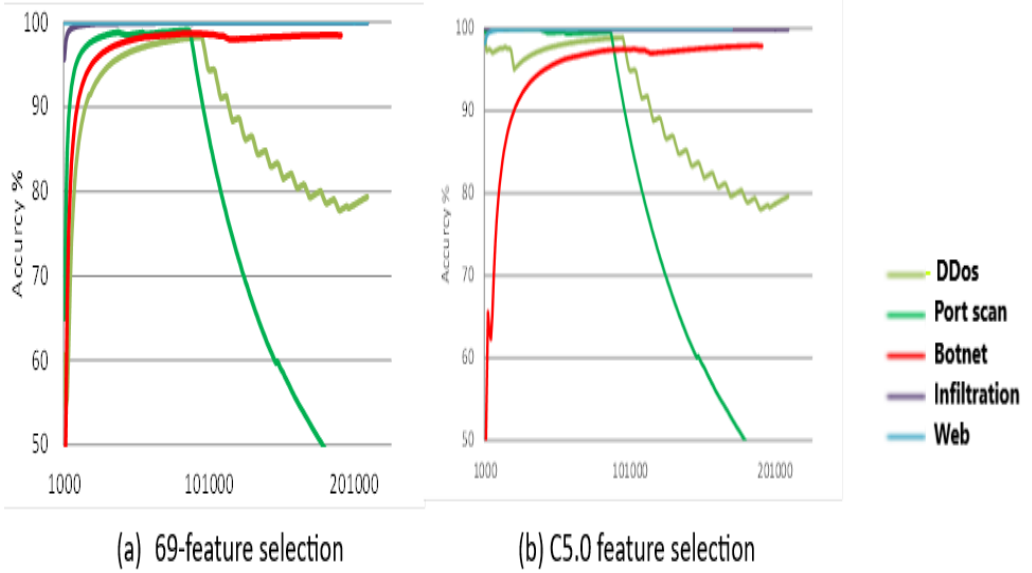

Figure 2-Comparison of cumulated accuracy in MLP with all CICIDS2017 datasets

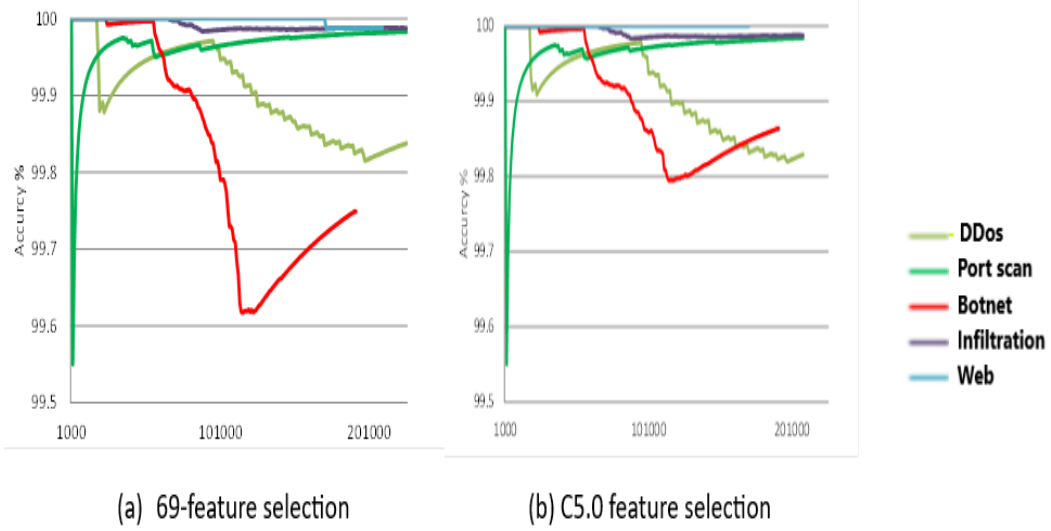

Figure 3-Comparison of cumulated accuracy in ARF with all CICIDS2017 datasets

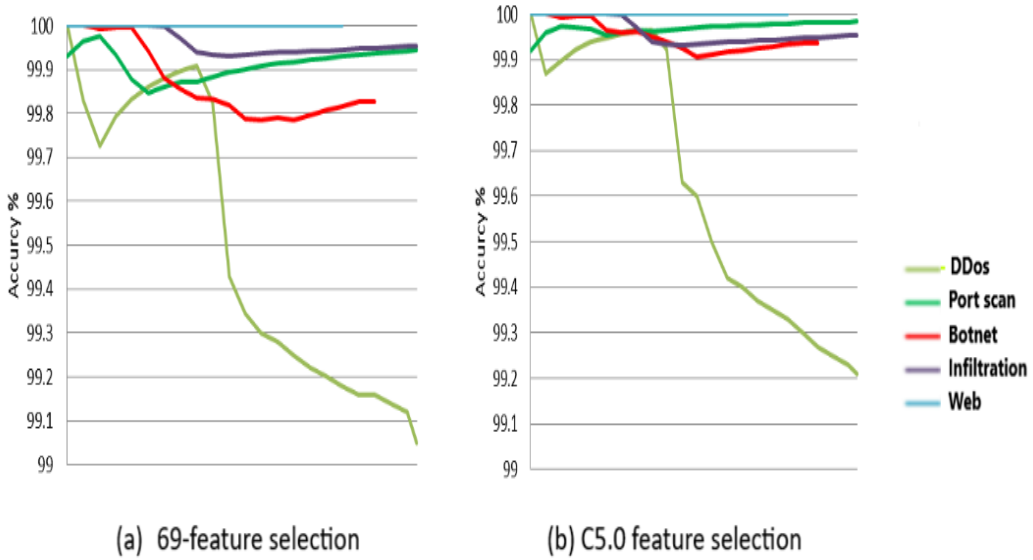

Figure 4-Comparison of cumulated accuracy in OZAboost with all CICIDS2017 datasets

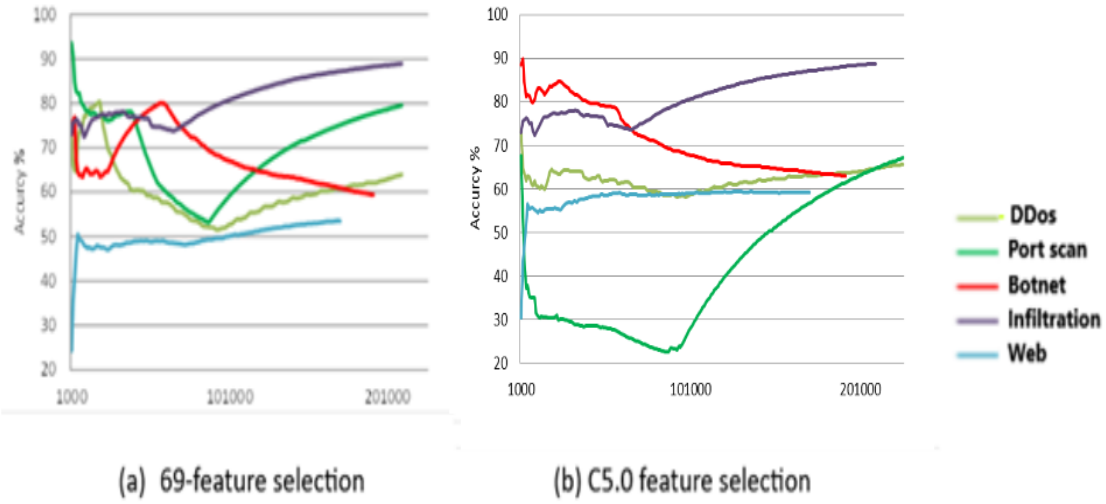

Figure 5-Comparison of cumulated accuracy in Naive-Bayes with all CICIDS2017 datasets

In general, Figures-(1, 2, 3, 4 and 5) show that, at all datasets with most of data stream algorithms used, the performance of the C5.0 feature selection was better than that of the 69 feature selection. On other hand, the results showed that even with less than 1% labeled data, the percentage of the total correct prediction on each segment for all datasets was approximately stable. For example, in Figure-1 (a), when SGD was used the values of change rate of cumulated accuracy were up to (±0.0027), (±0.0273), (±0.0089), (±0.0008) and (±0.0001) for DDoS, port-scan, botnet, Infiltration, and Web attack datasets, respectively, while Figure-1 (b) show that these values were up to (±0.0028), (±0.0134), (±0.0110), (±0.006) and 0 for the same datasets, respectively.

8. Conclusions

Learning using data streams is a comparatively new model. This paper presented an efficient intrusion detection system to prevent from new attacks based on data stream classification algorithm with additional incoming stream-based learning with limited label. The results indicated that it is possible to improve both the accuracy and computation time by selecting the highly important features for C5.0 algorithm. The performance can be further improved using different methods, such as preprocessing with data reduction and parameter tuning, that can improve the efficiency of the classifier.

References

1. Koch, R., Golling, M. G. and Rodosek, G. D. **2017**. Towards comparability of intrusion detection systems: New data sets. In Proceedings of the TERENA Networking Conference.
2. Sergio R., Bartosz K, Salvador G., Michał W and Francisco H. **2017**. A survey on data preprocessing for data stream mining: Current status and future directions, Neurocomputing,
3. Monika Ar and Chaitali Ch. **2017**. A Survey on Classification Algorithm for Real Time Data Streams using Ensembled Approach, International Conference on Multidisciplinary Research & Practice.
4. Akanksha M. Shrishrimal, Khwaja A and Syed A. **2017**. Increasing efficiency of Intrusion Detection System using Stream Data Mining Classification, IJARIIE-ISSN(O)-2395-4396, **3**(1).
5. Loo Hui Ru, Trias A. and Marsono N. **2014**. Online Data Stream Learning and Classification with Limited Labels , Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics, Yogyakarta, Indonesia.
6. Ireneusz C. and Piotr J. **2014**. Ensemble classifier for mining data streams, International Conference on Knowledge-Based and Intelligent Information & Engineering Systems
7. Saddam H. and Anirudh J. **2018**. Analysis of Network Intrusion Detection System with Machine Learning Algorithms (Deep Reinforcement Learning Algorithm). Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden
8. Iman Sh, Arash H. and Ali A. **2018**. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal,
9. Specht, S. and Lee R **2004**. Distributed denial of service: taxonomies of attacks, tools, and countermeasure, ISCA PDCS.
10. Shirey, R. **2000**. *Internet Security Glossary*, RFC Editor, United States.
11. Paul S. **2017**. Thingbots: The Future of Botnets in the Internet of Things, Security Intelligence.
12. Longadge, R., Dongre, S.S. and Malik, L. **2013**. Class Imbalance Problem in Data Mining: Review, *International Journal of Computer Science and Network* (IJCSN), **2**(1): 2277-5420.
13. Zliobaite I., Bifet A, Holmes G. and Pfahringer B. **2011**. *Moa conceptdrift active learning strategies for streaming data*, E-book.
14. Heito M., Jean P. and Fabrício E. **2017**. A Survey on Ensemble Learning for Data Stream Classification.
15. Heitor M., Albert B., Jesse R. and Jean P. **2017**. Adaptive random forests for evolving data stream classification.
16. Bottou L. **2010**. Large-Scale Machine Learning with Stochastic Gradient Descent, Proceedings of the 19th International Conference on Computational Statistics.
17. García S., Luengo J. and Herrera F. **2015**. *Data Preprocessing in Data Mining*, Springer.
18. García S., Luengo J. and Herrera F. **2016**. Tutorial on practical tips of the most influential data preprocessing algorithms in data mining, Knowl. Based Syst. 98.
19. Jundong L., Kewei Ch., Suhang W., Fred M., Robert P., Jiliang T. and Huan Lu. **2016**. Feature selection: A data perspective. arXiv preprint arXiv:1601.07996.