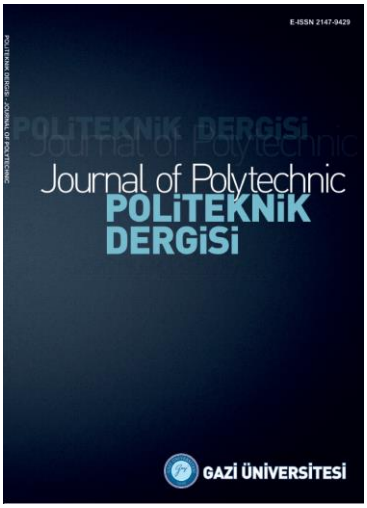

POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.org.tr/politeknik

Intrusion detection model based on TF.IDF and C4.5 algorithms

TF.IDF ve C4.5 algoritması tabanlı saldırı tesbit modeli

***Yazar(lar) (Author(s))**: Khaldoon AWADH[1], Ayhan AKBAŞ[2]*

ORCID[1]: *0000-0001-6697-931X*

ORCID[2]: *0000-0002-6425-104X*

Bu makaleye şu şekilde atıfta bulunabilirsiniz(To cite to this article): Awadh K., Akbas A., "Intrusion detection model based on TF.IDF and C4.5 algorithms", ***Politeknik Dergisi***, 24(4): 1691-1698, (2021).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: **10.2339/politeknik.693221**

Intrusion Detection Model based on TF.IDF and C4.5 Algorithms

Highlights

- *Employment of TF.IDF for transformation of network packet data for Machine Learning algorithms.*
- *An efficient model for distinguishing malicious network packets through the C4.5 algorithm.*
- *Remarkable improvements in the accuracy and performance with the combined use of TF.IDF and C4.5*

Graphical Abstract

In this study, an efficient model for the intrusion detection has been developed with TF.IDF and C4.5 algorithms. The model has been tested on a dataset for validation and results show a significant increase in accuracy in comparison with Multi Layer Perceptron (MLP) and Naive Bayes (NB).

Aim

In this study, the design of a new model has been aimed to increase accuracy of detecting malicious network packets for intrusion detection systems.

Design & Methodology

The model has been designed for increased efficiency and improved accuracy though the employment of TF.IDF and C4.5 algorithms.

Originality

TF.IDF and C4.5 algorithms have been used together to achieve efficient detection of malicious data.

Findings

TF.IDF algorithm is found to be very effective to transform network data into the form that Machine Learning algorithms can process. C4.5 algorithm outperforms compared to MLP and NB.

Conclusion

An efficient intrusion detection model design with the TF.IDF and C4.5 algorithms is introduced. Designed model has been compared with MLP and NB algorithms to verify the performance.

Declaration of Ethical Standards

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

TF.IDF ve C4.5 Algorithması Tabanlı Saldırı Tesbit Modeli

Araştırma Makalesi / Research Article

Khaldoon AWADH[1*], Ayhan AKBAŞ[2]

[1]Computer Engineering Department, University of Turkish Aeronautical Association, Turkey
[2]Computer Engineering Department, Cankiri Karatekin University, Turkey

(Geliş/Received : 24.02.2020 ; Kabul/Accepted : 04.07.2020 ; Erken Görünüm/Early View : 12.07.2020)

ÖZ

Son yıllarda, makine öğrenmesi ve veri madenciliği teknolojilerini kullanarak kullanarak Saldırı Tespit sistemlerinin performanslarını iyileştirmenin yeni yollarını keşfetmek araştırmacıların ilgisini çekmektedir. Bu konuda araştırmacıların karşılaştıkları en zorlayıcı problemlerden biri eldeki verilerin makine öğrenmesinde kullanabilecekleri forma dönüştürülmesidir. Bu makalede, simule edilmiş UNSW-NB15 veri setini dönüştürme ön işlemi ile beraber C4.5 algoritması karar ağacını temel alan Saldırı Tespit Sistemi modeli sunmaktadır. Oluşturulan model, yüksek tespit performansını yakalayabilmek amacıyla veri tiplerini, makine öğrenme işleminin kabul ettiği verimli bir forma dönüştürmek için Term Frequency-Inverse Document Frequency (TF.IDF) metodunu kullanmaktadır. Model, UNSW-NB15 veri setinin rastgele seçilmiş 250000 kaydı ile test edilmiştir. Seçilmiş kayıtlar, 50, 500, 1000, 5000 kayıtlık segmentler haline gruplandırılmıştır. Her segment daha sonra çoklu ve ikili sınıf veri setleri olarak alt gruplandırılmıştır. Weka yazılımında C4.5 karar ağacı algoritması ile Çoklu Katmanlı Perseptron (MLP) performansı ve Naïve Bayes ile karşılaştırılmıştır. Önerilen metod sınıflandırıcıların performansını ciddi oranda artırmış ve yanlış tahmin oranlarını düşürmüştür.

Anahtar Kelimeler: STS, TF.IDF, veri madenciliği, makine öğrenmesi, ağ güvenliği.

Intrusion Detection Model Based on TF.IDF and C4.5 Algorithms

ABSTRACT

In recent years, the use of machine learning and data mining technologies has drawn researchers' attention to new ways to improve the performance of Intrusion Detection Systems (IDS). These techniques have proven to be an effective method in distinguishing malicious network packets. One of the most challenging problems that researchers are faced with is the transformation of data into a form that can be handled effectively by Machine Learning Algorithms (MLA). In this paper, we present an IDS model based on the decision tree C4.5 algorithm with transforming simulated UNSW-NB15 dataset as a pre-processing operation. Our model uses Term Frequency.Inverse Document Frequency (TF.IDF) to convert data types to an acceptable and efficient form for machine learning to achieve high detection performance. The model has been tested with randomly selected 250000 records of the UNSW-NB15 dataset. Selected records have been grouped into various segment sizes, like 50, 500, 1000, and 5000 items. Each segment has been, further, grouped into two subsets of multi and binary class datasets. The performance of the Decision Tree C4.5 algorithm with Multilayer Perceptron (MLP) and Naive Bayes (NB) has been compared in Weka software. Our proposed method significantly has improved the accuracy of classifiers and decreased incorrectly detected instances. The increase in accuracy reflects the efficiency of transforming the dataset with TF.IDF of various segment sizes.

Keywords: IDS, TF.IDF, data mining, machine learning, network security.

1. INTRODUCTION

Today, computer networks have become a crucial element of all modern-day applications such as transportation, healthcare, banking, etc. The number of attacks and threats on networks to gain illegal access to sensitive data or resources is increasing enormously every day, along with the increasing amount of data generated. IDSs monitor and analyze network traffic behavior in the system against a potential intrusion that may violate the computer or network confidentiality, integrity, or availability [1]. These violations represent a direct threat to the security of the network. In the last thirty years, various researches have been done to implement IDS, using different techniques. However, none of these techniques is trustworthy, and cybercrime is still rising and retaining its potential [2]. Among all these approaches, we utilized data mining and machine learning algorithms on a public intrusion dataset to achieve higher detection rates [3]. We tested our proposed model with contemporary synthesized attack activities, using the UNSW-NB15 intrusion dataset [4]. This dataset is a modern one and more efficient than common datasets, namely, KDD98, KDDCUP99, and NSLKDD, due to its large number of features and a broader list of attack types. There are many studies using this dataset [5-12]. We used a supervised training schema of MLA as a proper strategy to solve intrusion detection

**Sorumlu Yazar (Corresponding Author)*
e-posta : aakbas@karatekin.edu.tr

problems owing to the huge volume of data. In our approach, we employ TF.IDF algorithm as a pre-processor. TF.IDF algorithm assigns a weight for every word in a document to find the special words that characterize documents as in search engines or document recognition applications for text mining. The converted dataset will be experimented and evaluated by using Decision Tree, Artificial Neural Networks, and Naive Bayes algorithms in classification mode.

2. RELATED WORK

This section presents related works in the literature on IDS tested with different types of datasets. Mehmood and Rais [13], for example, have experimented and presented a comparison of various supervised ML algorithms such as Support Vector Machine, Naive Bayes, J.48, and Decision Table with KDD99 dataset for anomaly detection. They used Weka 3.7 environment for their experiments. Their comparison showed that the accuracy of J.48 Decision Tree is the highest among all other algorithms and has a low misclassification rate. Mane and Pawar [14] proposed a model of anomaly IDS based on the ANN algorithm using only 17 attributes of 41 whole attributes in the KDD 99 dataset. Their results show that reducing the number of features increases the detection accuracy rate up to (98.0 %) and helps the training phase and testing phase to complete in a shorter time. Deshmukh, Ghorpade, and Padiya [15] depicted a comparative performance of different classification algorithms of Naive Bayes, AD Tree, and NB Tree by using benchmark NSL-KDD 99 dataset. They utilized Weka as a data mining tool for classification and analyzing the results. Their proposed model has improved the accuracy of classifiers detection rate for higher TP Rate of all the classifiers. Mogal, Ghungrad, and Bhusare [16], also proposed a model of IDS based on Central Points of attribute values with a prior algorithm to select high ranked features and remove irrelevant features because of the the fact that irrelevant features cause high FAR in the pre-processing stage before Logistic Regression and Naive Bayes algorithms are applied. The results revealed that the pre-processing reduced the processing time and improved the evaluation of the Naive Bayes algorithm. Another study by Dadgar S.M.H., M.S. Araghi, and M.M. Farahani [17] presented a novel text mining approach for classifying news text. Text pre-processing has been used due to the unstructured form of new text collected from different news resources to clean up useless items from the dataset. The weight of each word is calculated with the TF-IDF equation, which is one of the most famous text mining algorithms. The classification precision was 97.48 % and 94.93 % for the BBC and 20 Newsgroup datasets, respectively.

3. MODEL METHODOLOGY

In the UNSW-NB15 dataset, 49 features define the behavior of attack and regular records. UNSW-NB15 dataset consists of different data types such as integer, binary, float, timestamp, and nominal in addition to missing values. 250000 records are chosen randomly from the UNSW-NB15 dataset to experiment in the model. The portion of the selected dataset represents roughly 10 % of the total 2540044 of instances in the UNSW-NB15 dataset, which is available at the link (https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo4ys). The ratio of regular records equals 77.78 % of all records, and the attacker records equal to 22.22 % of records compared to overall records, both forming two attribute classes in the UNSW-NB15 dataset [5]. In this paper, two datasets have experimented with the proposed model. The first one is a multi-class dataset consisting of 10 classes: 9 classes of attack, namely, Fuzzers, Reconnaissance, Shellcode, Analysis, Backdoors, DoS, Worms, Exploits and Generic and one regular class. The second one is for the binary class dataset, which consists of either attack or normal.

3.1 Dataset Preprocessing

Data pre-processing improves the quality of data to make it more meaningful for classifiers by using data transformation and attributes reduction [18, 19]. It is the first step before the data manipulations. This process, ie. the sequences of TF.IDF processing steps is illustrated in Fig. 1.

3.2 TF.IDF Dataset Transformation

This research employs TF.IDF property with a structured UNSW-NB15 dataset form instead of unstructured text documents. TF.IDF Process converts the different data types of homogeneous data types and eliminates missing values by converting it into a uniform numeric form. The value of each item in the dataset will be replaced by the item's weight in the overall dataset. This model will run experiments with different datasets of various segment sizes, such as 50, 500, 1000, and 5000 records. Term Frequency (TF) is used to measure the occurrence of all terms repeated in a document with the consideration that the document is different in text length (number of words) [20]. $TF_{i,j}$ is the number of times a term *i* appears in a document j, and the denominator is the maxk number of a term that has the maximum occurrences from every term in the document *j* as in the following equation:

$$TF_{i,j} = \frac{F_{i,j}}{max_k F_{k,j}} \quad (1)$$

Inverse Document Frequency (IDF) measures the general importance of a term in an individual or several documents. IDF equation computes log_2 dividing the total number of documents in a collection *N+1* by *n* the number of documents that contain the term *i* [21]. One is added to the dividing values to prevent zero appearance in the dataset items.

$$IDF_i = log_2 \frac{N+1}{n_i} \quad (2)$$

Merging the $TF_{ij} \times IDF_i$ from equations 1 and 2 produces a composite weight for each term *i* in each document *j* [21]. The sequence of TF.IDF processing steps is:

1. Basically, to implement TF.IDF algorithm, a corpus C that contains a number M of documents D, is needed.

$$C = d_1, d_2, \ldots, d_M \quad (3)$$

In this work, the structured dataset that contains a number of records should be divided into several equal segment sizes *M* according to the number of records *R* in each segment, each segment will be considered as a document *r*, and overall new dataset will be considered as a corpus *C*', as below:

$$M = D \div R \quad (4)$$

$$C' = r_1, r_2, \ldots, r_M \quad (5)$$

2. Converting different data types, in addition to missing values in every record in every segment to string type as a first step in computing the weight of items.
3. Finding the item with the largest number of occurrences in a single segment (document) according to equation 1.
4. Computing TF of each item in a segment by using equation 1.
5. Computing IDF of each item in all segments by using equation 2.
6. The results of computing the TF.IDF of each item produces the numeric weight.
7. Replacing the generated value from the TF.IDF operation of each item with the original value of the same item.

3.3 Normalization

Normalizing values is a conversion method to limit dataset values by a higher and lower bound to prevent the impact of the large scale features over the lower scale features [3, 16]. TF.IDF presents normalization by TF Eqn. 1. For example, a segment of 100 records contains 470 items, and if we suppose an item occurs once and another item is repeated 100 times, then the TF of the first item equals to 1/100=0.01 and the TF for the second item equals to 100/100=1, so that:

$$0 < TF_{ij} \leq 1 \quad (6)$$

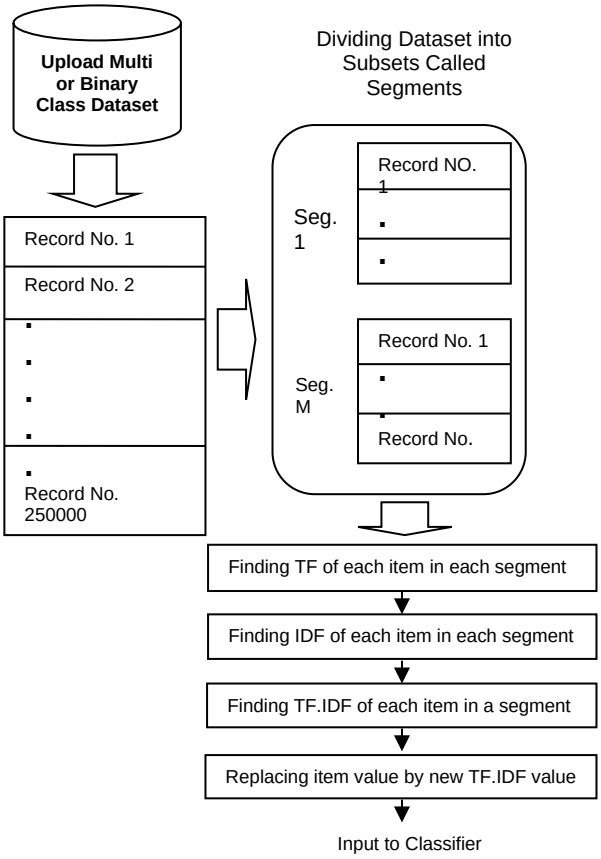

Figure 1. The sequences of TF.IDF processing steps

IDF part supports log_2 to minimize the rate of N over n_i as in Eqn. 2. For example, in this work, the 50-segment size dataset is the smallest segment size; therefore, it will have the largest number of segments; in this case, 5000 segments. If an item occurs only once in a document, then (5000+1/1)=5001, but with applying logarithm log_2(5000+1/1)=12.288, in the same case if the item occurs in all segments then log_2((5000+1)/5000)=0.0002 instead of 1.0002.

3.3.1 Attributes Reduction

UNSW-NB15 dataset has significant features that the authors of the dataset, Moustafa et al., presented in their study [15]. Our work relies on the way that they select the high ranked features of the UNSW-NB15 dataset in their work [15]. Since this work deals with the overall dataset, the features will be union to create a new set of appropriate attributes from the original dataset, with adding the multi or binary class attribute. The new subset will contain 30 attributes, which are, namely, Proto, state, dbytes, sttl, dttl, sloss, dloss, service, Sload, Dload, Spkts, Dpkts, swin, dwin, stcpb, smeansz, Djit, synack, ackdat, is_sm_ips_ports, ct_state_ttl, is_ftp_login, ct_ftp_cmd, ct_srv_src, ct_srv_dst, ct_src_ltm, ct_dst_ltm, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, with adding the multi or binary class attribute. Reducing attributes were applied after TF.IDF processing for wider flexibility in

attribute selection to our model. The main benefit of multi-class attack classification is that it gives us a thorough and deep understanding of the relationship between the dataset features and attack categories [22]. Likewise, for binary class, it is important to maintain the results of the classification within the circle of attack.

3.5 Machine Learning Classifiers

3.5.1 Decision Tree Algorithm C4.5

Decision Tree Algorithm C4.5 is also called as J48. J48 is an open-source algorithm coded in Java and available in the Weka software as a classification approach. C4.5 is an enhanced version of the Decision Tree Algorithm ID3 developed by Quinlan in 1993 [22]. It is a supervised classification algorithm [23]. C4.5 uses information gain and entropy to choose and splitting the set of attributes from the multiple attributes dataset as a subset of one class or more [20]. Pruning is used to increase the classifier accuracy for noisy data or in classifying instances that do not belong to the class predicted by that leaf [22, 23, 24]. The pruning mechanism starts after the decision tree has been created. It checks the tree nodes and attempts to mitigate branches by replacing undesired nodes by leaf nodes [25].

3.5.2 Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) is a multi-layer feedforward neural network. It is one of the successful classification algorithms in the IDS realm. Since it is a feedforward, the data is transmitted in one way from input to output by hidden layer(s) [20]. Afterward, data is transferred to the next layer. The sigmoid non-linear activation function is applied in each node. Backpropagation is used to predict the weights in the first network layers by error estimation from the errors in subsequent layers to update the weight of the node after computing the error with the consideration of the weight of that node [20].

3.5.3 Naive Bayes

Naive Bayes predicts the result of classification by using the conditional probabilities of Bayes theorem. Several applications use Bayes theorem in probability and statistics. The naive Bayes classifier model is a simplified Bayes classifier model [21], which can invert the posterior probability in the form of likelihood with class and predictor prior probability.

3.6 Validation

The cross-validation has been made for machine learning performance testing and evaluating the effectiveness [26]. It tests the whole dataset in K times by testing one part at a time. 10 folds cross-validation with both multi and binary class datasets are picked for classifiers to test the model performance.

4. RESULTS AND DISCUSSION

The performance of classifiers that mentioned previously was experimented with using Weka 3.8 environment.

4.1 IDS Model Performance Evaluation

The performance of the Intrusion Detection Model has been evaluated to find the detection and failing detection rate in the model [22, 23, 24, 25, 26, 27, 28]. The confusion matrix of the two-dimensional table displays the performance results of ML classification. TP=True Positive, FP=False Positive, TN=True Negative, and FN=False Negative are four measurement principles that can be generated from the confusion matrix [21, 23, 24, 25, 26]. ML measurements such as True Positive Rate (TPR), False Positives Rate (FPR), Precision, Accuracy, Error Rate, and F-measure can be concluded from the measurement principles. True Positive Rate (TPR) is known as sensitive [21]; it is the number of instances in the dataset that are classified correctly for all classes.

$$TPR = \frac{TP}{TP+FN} \times 100 \quad (7)$$

False positives rate (FPR) represents the number of instances for all classes in the dataset that are classified incorrectly [21].

$$FPR = \frac{FP}{FP+TN} \times 100 \quad (8)$$

Precision represents the measure of the probability that positive instances are predicted correctly [28].

$$precision = \frac{TP}{TP+FP} \times 100 \quad (9)$$

Accuracy is the percentage of instances that are detected correctly over all the instances in the dataset.

$$accuracy = \frac{TP+TN}{n} \times 100, \quad n = TP + TN + FP + FN \quad (10)$$

Error Rate is known as incorrectly classified instances. It is the percentage of the incorrectly detected instances over all the instances of the dataset [29].

$$errorrate = \frac{FP+FN}{n} \times 100, \; n = TP + TN + FP + FN \quad (11)$$

F-measure scores the balance between precision and TPR. The F-measure is considered as the harmonic mean of TPR and precision [30].

$$f_{measure} = \frac{2 \times precision \times TPR}{precision + TPR} \quad (12)$$

The above equations are the global measures of ML classifier performance. The classifier gets more efficient and reliable as the accuracy rate comes closer to 100 %, and the error rate closer to 0 %. The Weka software provides all indicators mentioned above.

4.2 Results of the Experiments

This part presents the results of implementations of MLP, C4.5, and NB classifiers with multi and binary class pre-processed datasets. The results of the multi-class datasets appear in Tables 1, 2, and 3, and the results for binary-class datasets appear in Tables 4, 5, and 6

4.3 Results Discussion

Experiments of both multi-class datasets and the binary class dataset were carried out with 50, 500, 1000, and 5000 records of segment size. Different segment sizes in both binary and multi-class IDS datasets produced different accuracies; Tables 1 to 6 show the changes in the accuracy of every classifier for each dataset. This change in accuracy reflects the efficiency of transforming the dataset with TF.IDF with various segment sizes. C4.5 classifier achieved the highest accuracy, highest TPR, and the lowest FPR with the UNSW-NB15 multi-class dataset. The accuracy was 97.36 %, the TPR was 97.40 %, and FPR was 1.3 %. Fig. 2 illustrates the comparison between C4.5 and other classifiers with multi class datasets. C4.5 classifier achieved the highest accuracy, highest TPR, and the lowest FPR with the UNSW-NB15 binary class dataset. The accuracy was 99.43 %, the TPR was 99.4 %, and FPR was 1.1 %. Fig. 3 shows a comparison between C4.5 and other classifiers with binary class datasets. Naive Bayes algorithm was distinguished from the other classifiers in building model time. It consumed the lowest time between 3.2 seconds and 1.5 seconds, but the model performance has dropped down. It achieved the lowest accuracies with an effective detection ratio of classes.

Comparatively, MLP has performed with reasonable accuracy on the dataset with a segment size of 5000-records. Still, it was unable to reach to C4.5 in terms of accuracy and number of detecting classes. MLP was able to detect between 6 and 7 classes from 10. None of the experiments was able to detect ten classes of multi datasets, as illustrated in Fig. 4. Some attack types rarely appear in the UNSW-NB15 dataset. For example, Worms attacks appear 174 times in the overall dataset of 250044 records. In our randomly selected dataset, which contains 250000 records, there are only 19 records of Worms attacking class, which can be described as an imbalanced dataset. The property of the similarities between the values of UNSW-NB15 dataset records showed that the ML could not detect several records categories [27]. This property has been reduced by using TF.IDF for the transformation of dataset records. Evidence of this effect is the high accuracy detection results.

5. CONCLUSION

In this work, we developed an Intrusion Detection Model based on decision tree algorithm C4.5 and TF.IDF. The model was tested with various datasets, each of which has a specific segment size of 50, 500, 1000, and 5000 records from 250000 randomly selected records. The dataset is characterized as multi and binary class datasets.

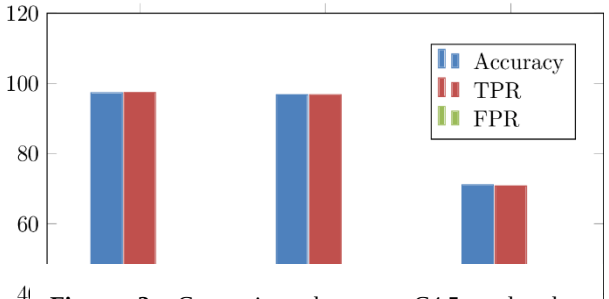

Figure 2. Comparison between C4.5 and other classifiers with multi-class datasets.

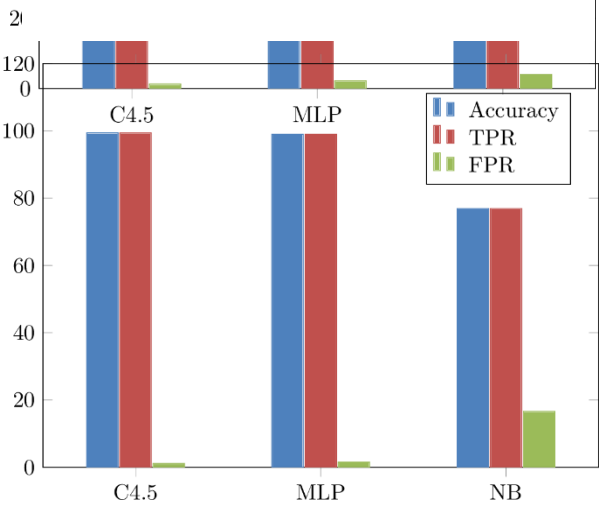

Figure 3. Comparison between C4.5 and other classifiers with binary class datasets

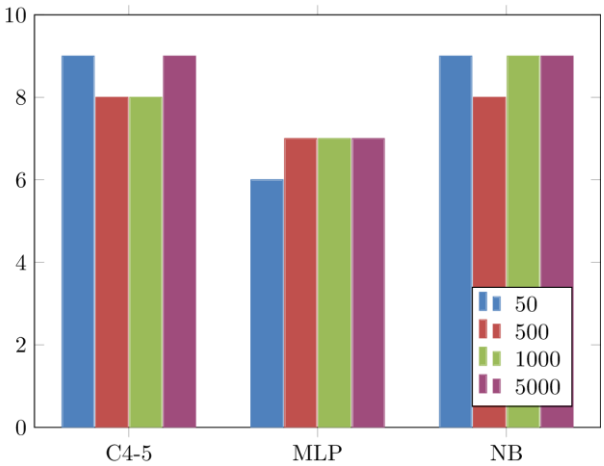

Figure 4. Classifiers with a different segmented dataset (out of 10 classes)

Table 1. The performance of the C4.5 algorithm wit h multi-class datasets.

Seg.Size	Acc.	ER.	TPR	FPR	Pre.	F-M	Time Sec.	Class No
50	96.79 %	3.20 %	0.968	0.021	0.966	0.966	91	9
500	97.27 %	2.73 %	0.973	0.013	0.971	0.971	32	8
1000	97.36 %	2.63 %	0.974	0.013	0.972	0.972	36	8
5000	97.36 %	2.61 %	0.974	0.013	0.972	0.972	92	9

Table 2. The performance of MLP with multi-class datasets.

Seg.Size	Acc.	ER.	TPR	FPR	Pre.	F-M	Time Sec.	Class No
50	96.46 %	3.54 %	0.965	0.027	0.963	0.964	4563	6
500	96.63 %	3.37 %	0.966	0.025	0.965	0.965	4958	7
1000	96.73 %	3.27 %	0.967	0.022	0.966	0.966	5018	7
5000	96.78 %	3.22 %	0.968	0.022	0.966	0.967	3990	7

Table 3. The performance of NB with multi-class datasets.

Seg.Size	Acc.	ER.	TPR	FPR	Pre.	F-M	Time Sec.	Class No
50	71.10 %	28.90 %	0.711	0.040	0.861	0.747	1.64	9
500	70.80 %	29.20 %	0.708	0.040	0.860	0.746	2.17	8
1000	70.33 %	29.67 %	0.703	0.039	0.859	0.743	1.71	9
5000	70.47 %	29.63 %	0.705	0.039	0.859	0.744	2.44	9

Table 4. The performance of C4.5 with binary class datasets.

Seg.Size	Acc.	ER.	TPR	FPR	Pre.	F-M	Time Sec.	Class No
50	98.98 %	1.02 %	0.99	0.019	0.99	0.99	115	2
500	99.36 %	0.64 %	0.994	0.012	0.994	0.994	85	2
1000	99.43 %	0.57 %	0.994	0.011	0.994	0.994	153	2
5000	99.35 %	0.65 %	0.994	0.013	0.994	0.994	186	2

Table 5. The performance of MLP with binary class datasets.

Seg.Size	Acc.	ER.	TPR	FPR	Pre.	F-M	Time Sec.	Class No
50	98.64 %	1.36 %	0.986	0.022	0.987	0.986	4221	2
500	98.92 %	1.08 %	0.989	0.016	0.989	0.989	3582	2
1000	98.96 %	1.04 %	0.99	0.018	0.99	0.99	4229	2
5000	99.00 %	1.00 %	0.99	0.015	0.99	0.99	4237	2

Table 6. The Performance of NB with binary class datasets.

Seg.Size	Acc.	ER.	TPR	FPR	Pre.	F-M	Time Sec.	Class No
50	76.65 %	23.35 %	0.766	0.165	0.846	0.785	1.5	2
500	76.88 %	23.12 %	0.769	0.166	0.847	0.787	1.6	2
1000	76.89 %	23.11 %	0.769	0.166	0.847	0.787	1.5	2
5000	76.88 %	23.12 %	0.769	0.165	0.847	0.787	3.2	2

Only 30 out of 47 attributes from the UNSW-NB15 dataset were exploited. We have compared the performance of Multilayer Perceptron and Naive Bayes as various classification techniques with decision tree algorithm C4.5 by using Weka software with 10 fold cross-validation. From the results of detection indicators we concluded that the C4.5 classifier achieved the highest accuracy 97.36 %, highest TPR 97.40 % and the lowest FPR 1.3 % with 5000 segment size of the multi-class dataset, with the same classifier, but with 1000 segment size of the binary class dataset the highest accuracy 99.43 %, highest TPR 99.4 %, and the lowest

Table 7. The increase in accuracy of every classifier

ML Dataset	Class Type	Highest	Seg. Size	Lowest	Seg. Size	Inc.
2*C4.5	Multi	97.36 %	5000	96.80 %	50	0.56 %
	Binary	99.43 %	1000	98.98 %	50	0.45 %
2*MLP	Multi	96.78 %	5000	96.46 %	50	0.32 %
	Binary	99.00 %	5000	98.64 %	50	0.36 %
2*NB	Multi	71.10 %	50	70.33 %	1000	0.77 %
	Binary	76.89 %	1000	76.65 %	50	0.24 %

FPR 1.1 %. Different segment sizes increase the accuracy of C4.5 in both binaries to 0.6 % and multi-class to 0.45 %, which leads to yield different accuracies. TF.IDF as a transformer increased the effectiveness of the C4.5 algorithm since it deals with continuous value attributes and mitigates the overfitting of the data by pruning. For future work, it would be interesting to test different Machine Learning Algorithms such as Random Forest or Random Tree with different segment sizes with the UNSW-NB15 dataset.

DECLARATION OF ETHICAL STANDARDS

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

AUTHORS' CONTRIBUTIONS

Khaldoon AWADH: Performed the experiments and analyse the results.

Ayhan AKBAŞ: Performed the experiments and analyse the results.

CONFLICT OF INTEREST

There is no conflict of interest in this study.

REFERENCES

[1 Yu Z.,"Intrusion Detection: A Machine Learning Approach". ***Imperial College Press London, UK*** (2011).

[2] Armin J, Thompson B, Ariu D, Giacinto G, Roli F, Kijewski P.,"Cybercrime economic costs: No measure, no solution". ***10th International Conference on Availability, Reliability and Security (ARES),*** Toulouse, France, 701-710 (2015).

[3] Bhattacharyya D.K., Kalita J.K., "Network anomaly detection: A machine learning perspective". ***New York, USA: CRC Press***, (2013).

[4] Moustafa N., Slay J., " UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)". ***IEEE 2015 Military Communications and Information Systems Conference (MilCIS)***, Canberra, Australia, 1-6, (2015).

[5] Moustafa N., Turnbull B., Choo K. R. , " An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things.", ***IEEE Internet of Things Journal*** , 4815-4830 ,(2018).

[6] Koroniotis N., Moustafa N., Sitnikova E., Slay J. "Towards Developing Network Forensic Mechanism for Botnet Activities in the IoT Based on Machine Learning Techniques." ***International Conference on Mobile Networks and Management. Springer, Cham***, (2017).

[7] Moustafa N., Misra G., Slay J., "Generalized Outlier Gaussian Mixture technique based on Automated Association Features for Simulating and Detecting Web Application Attacks.", ***IEEE Transactions on Sustainable Computing***, 1-1, (2018).

[8] Keshk M., Moustafa N., Sitnikova E., Creech G., "Privacy preservation intrusion detection technique for SCADA systems.", ***Military Communications and Information Systems Conference (MilCIS)***, Canberra, 1-6 (2017).

[9] Moustafa N., Adi E., Turnbull B., Hu J.,"A New Threat Intelligence Scheme for Safeguarding Industry 4.0 Systems.", 32910-32924, ***IEEE Access***, (2018).

[10] Moustafa N., Creech G., Slay J., "Anomaly Detection System Using Beta Mixture Models and Outlier Detection." ***Progress in Computing, Analytics and Networking. Springer,*** 125-135, (2018).

[11] Moustafa N., Creech G., Slay J., "Flow Aggregator Module for Analysing Network Traffic." ***Progress in Computing, Analytics and Networking. Springer,*** 19-29, (2017).

[12] Moustafa, N., Slay J., "A Network Forensic Scheme Using Correntropy-Variation for Attack Detection." ***IFIP International Conference on Digital Forensics. Springer***, 225-239, (2018).

[13] Mehmood T, Rais H., "Machine learning algorithms in context of intrusion detection". ***IEEE 2016 Computer and Information Sciences (ICCOINS)***, Kuala Lumpur, 369-373, (2016).

[14] Mane D., Pawar S., "Anomaly based IDS using Backpropagation Neural Network", ***International Journal of Computer Applications***; 29-34,(2016).

[15] Deshmukh D. H., Ghorpade T., Padiya P., "Intrusion detection system by improved pre-processing methods and Naive Bayes classifier using NSL-KDD 99 Dataset", ***IEEE 2014 International Conference on Electronics***

and Communication Systems (ICECS), Coimbatore,1-7, (2014).

[16] Mogal D. G., Ghungrad S.R., Bhusare B.B., "NIDS using Machine Learning Classifiers on UNSW-NB15 and KDDCUP99 Datasets". ***International Journal of Advanced Research in Computer and Communication Engineering,*** 533-537, (2017).

[17] Dadgar S. M. H., Araghi M. S., Farahani M. M.,"A novel text mining approach based on TF-IDF and Support Vector Machine for news classification". ***IEEE 2016 International Conference on Engineering and Technology (ICETECH)***; Coimbatore, India,112-116. (2016).

[18] Leskovec J., Rajaraman A., Ullman J. D., "Mining of massive datasets. 5th ed.", ***England: Cambridge University Press***, (2014).

[19] Schütze H., Raghavan P., Manning C. D., "An Introduction to Information Retrieval", ***England, Cambridge University Press***, (2009).

[20] Aggarwal C. C., "Data mining: the textbook", ***New York, USA: Springer***, (2015).

[21] Zaki M. J., Meira W., "Data mining and analysis: fundamental concepts and algorithms", ***New York, USA: Cambridge University Press***, (2014).

[22] Hssina B., Merbouha A., Ezzikouri H., Erritali M., "A comparative study of decision tree ID3 and C4.5***". International Journal of Advanced Computer Science and Applications,*** 13-19, (2014).

[23] Witten I. H., Frank E., Hall M. A., Pal C. J., "Data Mining: Practical machine learning tools and techniques. 4th ed.", ***Morgan Kaufmann***, (2016).

[24] Kumar V., Wu X. "The Top Ten Algorithms in Data Mining: Data Mining and Knowledge Discovery", ***Chapman and Hall/CRC***, (2009).

[25] Revathy R., Lawrance L., "Comparative Analysis of C4.5 and C5.0 Algorithms on Crop Pest Data", ***International Journal of Innovative Research in Computer and Communication Engineering,***50-58, (2017).

[26] Choudhury S., Bhowal A., "Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection", ***IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)***; Chennai, IndiaUSA, 89-95, (2015).

[27] Moustafa N., Slay J., "The significant features of the UNSW-NB15 and the KDD99 data sets for Network Intrusion Detection Systems", ***IEEE 4th Intl Workshop on Building Analysis Datasets and Gathering Experience Returns for Security,*** Kyoto, Japan, 25-31, (2015).

[28] Garg T., Khurana S. S., "Comparison of classification techniques for intrusion detection dataset using WEKA", ***IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014),*** Jaipur, India, 1-5, (2014).

[29] Gadal S. M., Mokhtar R., "Anomaly detection approach using hybrid algorithm of data mining technique", ***IEEE International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE),*** Khartoum, Sudan, 1-7, (2017).

[30] Elhamahmy M. E., Elmahdy H. N., Saroit I. A., "A new approach for evaluating intrusion detection system", ***International Journal of Artificial Intelligent Systems and Machine Learning***, 2: 290-298, (2010).