TAF

Journal of Applied and Physical Sciences

JAPS

2019, **5**(1): 30-36

PRIMARY RESEARCH

A distributed intrusion detection system based on apache spark and scikit-learn library

Mohamed Seghire Othman Djediden [1*], Hicham Reguieg [2], Zoulikha Mekkakia Maaza [3]

[1, 2, 3] Laboratoire SIMPA, Universite des Sciences et de la Technologie d'Oran Mohamed Boudiaf, USTO-MB, Oran, Algeria

Keywords

Intrusion detection
Machine learning
Big data
Distributed computing
Apache spark
Scikit-learn

Received: 12 January 2019
Accepted: 12 February 2019
Published: 28 February 2019

Abstract

With the great explosion of data generated in computer networks. The main task of Intrusion Detection Systems (IDS) has become more complicated. Most of the existing IDS are deployed on a single server and do not support the distributed processing. These systems encountered several problems as soon as the volume of the data to be analysed is larger and more varied. The main goal of this paper is to create an intrusion detection system that can analyse massive data quickly with great precision while supporting distributed data processing. This type of data processing assures that our system will be more available and fault-tolerant. In our work, we have combined the Apache Spark framework with known feature selection methods and machine learning algorithms from the improved Sickit-learn library called Sk-dist. The UNSW-NB15 dataset was used to assess the performance of our system. The results of comparisons made with other existing work have shown that our approach is much better in terms of accuracy, reduction of features and above all fault tolerance.

I. INTRODUCTION

IDS is designed to protect information systems against intrusion, duplication, misuse, destruction, and alteration. There are two types of IDS: (i) Signature-based which can detect known attacks using a database of signatures. This database is compared with the system activities. The advantage of this type is to avoid the generation of a high number of false alarms but it cannot detect unknown and zero-day attacks. (ii) Anomaly-based: This type detects unknown attacks in network traffic. They use machine-learning techniques to create normal network behaviour and each deviation from this behaviour will be identified as an anomaly. Compared to the first type, it allows to detect unknown and zero-day attacks but it suffers from a high false-positive rate because each traffic different from the created behaviour will be classified as an anomaly even if it is legitimate [1, 2].
One of the most used machine learning libraries in the creation of Anomaly-based IDS is the Scikit-learn library. Based on python this library provides much functionality like Classification, regression, Clustering, Model selection and preprocessing. The limitation of this library is that it does not support parallelization means that a Scikit-learn program cannot be run and distributed on a cluster [3].
Nowadays the number of users of computer networks (internet) has exploded; the captured network traffic has become more varied and more voluminous (Big Data). This is why the detection of intrusion using the traditional tools and methods is a very difficult and complicated task. Recent research aims to introduce Big Data analysis techniques and tools in the creation of IDS. The major challenges in this area is to create fault-tolerant distributed IDS that can analyse a fairly large and varied set of data while ensuring better accuracy. In this paper, we use Apache Spark as a data processing framework.
This fast Big Data framework covers different workloads like iterative algorithms, batch applications, streaming and interactive queries [4, 5, 6, 7].
The choice of the dataset used to test the proposed IDS is

*Corresponding author: Mohamed Seghire Othman Djediden
†email: mohamed.djediden@univ-usto.dz

very essential. For our approach, we opted for the UNSW-NB15 dataset as research has shown that UNSW-NB15 is more complex than other datasets and it better represents the current networks [8].

In this paper, we aim to get around the limitation of the undistributed Scikit-learn library by integrating a new package called Sk-dist, this python package built on top of Scikit-learn can be defined as an optimized version of Scikit-learn which supports distribution and parallelization over spark cluster. This allows distributed training of two classifiers: random forest and extra tree without any constraint on the physical resources [9]. In addition, to ensure the efficiency of our IDS, our approach combines the random forest classifier of this package with feature selection methods (chi-square and correlation-based) and the Spark framework. The IDS created selects the best subset of features ensuring higher accuracy. The main objective is to overcome the major limitation of the Sk-learn library (non-distributed processing) to benefit from the multiple-choice of these algorithms in terms of classification and feature selection. Because the choice is very limited in the event of use of machine-learning library integrated into Spark (Spark-ML). The Proposed approach will be tested and evaluated on the UNSWNB15 dataset.

For this purpose, this paper is organized as follows. In section 2, we introduce some related works on the application of the traditional tools and libraries (Scikit-learn, Weka and MATLAB) for IDS. In section 3, each step in the proposed approach is described. Section 4 presents the proposed approach results. Finally, we conclude our work and describe future work in section 5.

II. RELATED WORKS

In this section, we present several works that have created and developed IDS by applying machine learning algorithms and feature selection methods in undistributed environments with traditional tools like MATLAB, WEKA, and Scikit-learn. All the works cited used the UNSW-NB15 data set to test the effectiveness of their approaches.

Anwer et al. [10] combine two ML classifiers (Decision Tree (DT) and Naive Bayes (NB)) and different FS strategies to select the minimum number of features ensuring better accuracy. The authors used WEKA as a development environment and UNSW-NB15 to test their approach. The results prove that the best combination is the use of the Gain Ratio (GR) selector method and DT J48 as a classifier. Divekar et al. [11] Aim to optimize the stage of data pre-processing using the SMOTE oversampling and the random undersampling technique [12]. At the end of this stage, they combine different ML classifiers (Neural Network (NN), Support Vector Machine (SVM), DT, Random Forest (RF), NB, and K-Means) and selection methods from the Scikit-learn library to finalize their approach and to get the best accuracy. The KddCup99, NSL-KDD, and UNSW-NB15 datasets are used for the evaluation of the approach. The results obtained show that UNSW-NB15 can substitute the archaic KDD CUP 99 dataset and even NSL-KDD.

[13] use Random Forest as an ML classifier and Weka framework to improve intrusion detection. The authors propose a new dataset (with some features) from the original UNSWNB15 dataset and then compare this subset with the previous work in the KDD'99 dataset. The new subset shows better intrusion detection rates. The authors of [14] use different FS algorithm (genetic algorithm-logistic regression (GALR)) with different ML classifier (DT, RF, and NB) to classify the KDD99 and UNSW-NB15 datasets with the new optimal subset of features. The main objective of the authors is to obtain a better accuracy with a minimum number of features. The approach is developed on Weka and the results show that UNSW-NB15 it better represents the current networks over KddCup99. In [15], authors have created a new model containing two essential stages. The first uses a probability score value to classify network traffic as normal or abnormal (binary classification). The results of this stage are used as an additional feature in the final stage (multi-classification). The approach is developed using MATLAB and the KDD99 and UNSWNB15 datasets are used for experimentation. Results prove that the novel approach achieving high recognition rates.

[16] create a new hybrid features selection method, based on the Central Points (CP) of attribute values and Association Rule Mining (ARM). The proposed approach reduces the processing time overall by selecting the most frequent values and removing irrelevant or noisy features. The authors opted for Visual Studio C# 2008 as a development environment and have chosen Expectation-Maximization (EM) clustering, Logistic Regression (LR) and NB classifiers as the classification algorithm. This approach is applied to UNSW NB15 and NSL-KDD datasets. Results prove an improvement in accuracy and processing time. [17] analyse the UNSW-NB15 complexity according to three aspects of the statistical analysis phase, the feature correlation phase, and the complexity evaluation phase. The ML algorithms (DT, LR, NB, Artificial Neural Networks (ANN) and EM clustering) are used in the third phase to measure the complexity in terms of accuracy and (FAR) of UNSW-NB15 then the authors compared the results with the KDD99 dataset, the programming environment is Visual Studio Business Intelligence 2008. This paper proves that UNSW-NB15 is the best dataset to use for evaluating IDS efficiency.

ISSN: 2414-3103
DOI: 10.20474/japs-5.1.4

TABLE 1
UNSW-NB15 RELATED WORK COMPARATIVE

Reference	FS	ML Algorithm	ML (Best result)	Tools
[10]	Different filter and wrapper	DT and NB	DT	WEKA
[11]	Gini Impurity Index	NN, SVM, DT, RF, NB and K-Means.	RF	Scikit-learn
[13]	most frequently appeared features in intrusion	RF	RF	WEKA
[14]	GALR	DT,RF and NB	DT	WEKA
[15]	-	Deep learning method		MATLAB
[16]	CP and ARM	EM clustering , LR and NB	LR	Visual studio C 2008
[17]	No	NB, DT, ANN, LR, and EM clustering	DT	Visual Studio Business Intelligence 2008

Table 1 summarizes the different characteristics of the studied works. All the research cited in Table 1 uses tools and frameworks (environment) which cannot be run on a cluster and which does not support fault tolerance, scalability, high availability, distribution, and parallelization. These approaches cannot handle large and varied data sets, so since these approaches are undistributed, as soon as the server where they are developed will be down, the intrusion detection system will be stopped and the networks will be vulnerable to attacks.

To deal with all these limitations, we integrate a new package called Sk-Dist that supports distribution and parallelization over spark cluster. Our proposed approach combines the random forest classifier of this package with feature selection methods (chi-square and correlation-based) and the Spark framework and is evaluated with the UNSW-NB15 dataset. The IDS created selects the best subset of features ensuring higher accuracy. The approach will be detailed in the next section.

III. PROPOSED APPROACH

This section explains in detail our proposed approach by defining the techniques and the tools used to create our IDS. Figure 1 presents the workflow of our approach.

A. *Data Loading and Pre-Processing*

In our paper we opted for the UNSW-NB15 data set because it better represents current computer networks, this data set is composed of two partitions (training and testing dataset) and each partition contains 43 features plus two labels: ”label” which equals zero for normal traffic and one for attacks,” attack cat” which lists nine categories of attacks existing in UNSW-NB15.

We start by loading the two UNSWNB15 datasets into Spark using the python library named pandas. Among the 43 features of UNSW-NB15, there are three categorical features (’service’, ’proto’ and ’state’) which pose a problem for ML classifiers during training and prediction.

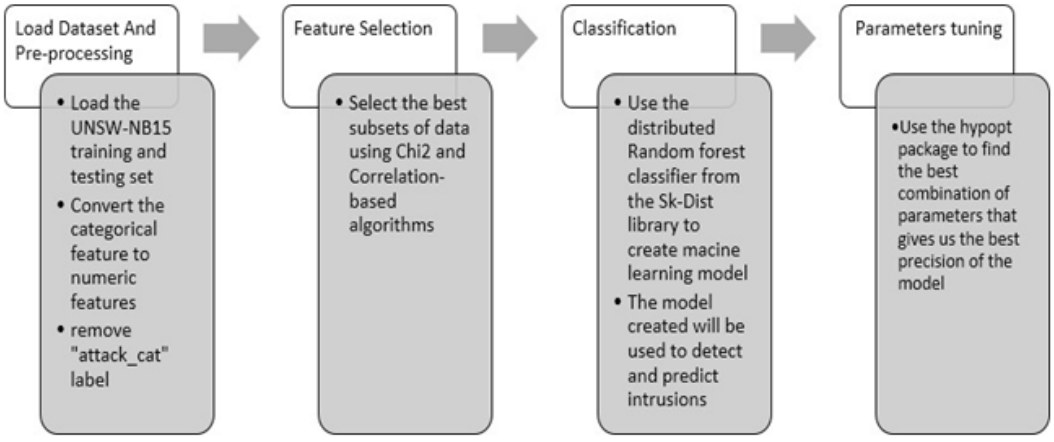

Fig. 1. Proposed approach workflow

To solve this problem we used the Label encoder function (from the Sklearn library) to convert these features to numeric features. Finally, since we aim in our approach to detect only attacks without citing their type, we removed the attack cat attribute from the two data sets (training and testing).

ISSN: 2414-3103
DOI: 10.20474/japs-5.1.4

Feature Selection

The Sklearn library offers several methods for the selection of features, for our IDS we tested the two selectors known for their effectiveness (chi-square and Correlation Based).
The Chi2 method computes chi-squared stats between each feature and the label. The calculation results are then used in the SelectKBest function to select only the features with the maximum chi-squared values [18]. The second Correlation-Based method (CB) calculates the absolute value of the Pearsons correlation between the features and the label and then keeps only the top n features based on this criterion.

Distributed ML Classifier

All existing classifiers in the Sklearn library cannot be executed on a cluster (undistributed classifiers) and do not support parallelization, this is why we integrated the Sk-Dist library in our approach, this library contains two ML classifiers (random forest and Extra tree) which allows you to create a shared and distributed ML model in a Spark cluster.
We chose random forest as the classifier because all the works cited above have proven their effectiveness for binary classification.

Parameter Tuning

The random forest algorithm in sk-dist has several parameters; the choice of values for these parameters directly influences the results of the classification (accuracy of the model). To find the best combination of parameters that gives us the best accuracy, we opted for the Hypopt package. This python package designed to optimize the parameters of ML algorithms is known by the use of a validation set (different from the package cross-validation of the Sklearn library that does not support this option). Also, this package makes the process of obtaining the best combination faster by supporting distribution in a cluster [18].
To assess the performance of our approach, two metrics are used: accuracy and f1-weighted score. These two metrics will be used in the next section to make comparisons with the other existing works. Here are the calculation formulas for the two metrics used:
Accuracy $= (TP + TN)(TP + TN + FP + FN)$
Where: $TP =$ Irue positive
$FP =$ False posithe
$TN =$ True negative and
$FN =$ False negative

$$F1 - \text{Weighted} = \frac{\sum_{i=1}^{K} \text{Support}_i . F1_i}{\text{Total}}$$

Where F1i is the F1-Score predicted for the ith target class.

$$F1 - Score = 2 * \left(\frac{recall \times precision}{recall + prectsion} \right)$$

IV. RESULTS AND DISCUSSION

In this section, we present the results of our proposed approach. In addition, the comparison made with the other existing works will be discussed to prove that our IDS ensures a better accuracy with a reduced number of features. In Table 2 and Figure 2 we compare our results with [9] using the f1-weighted score metric.

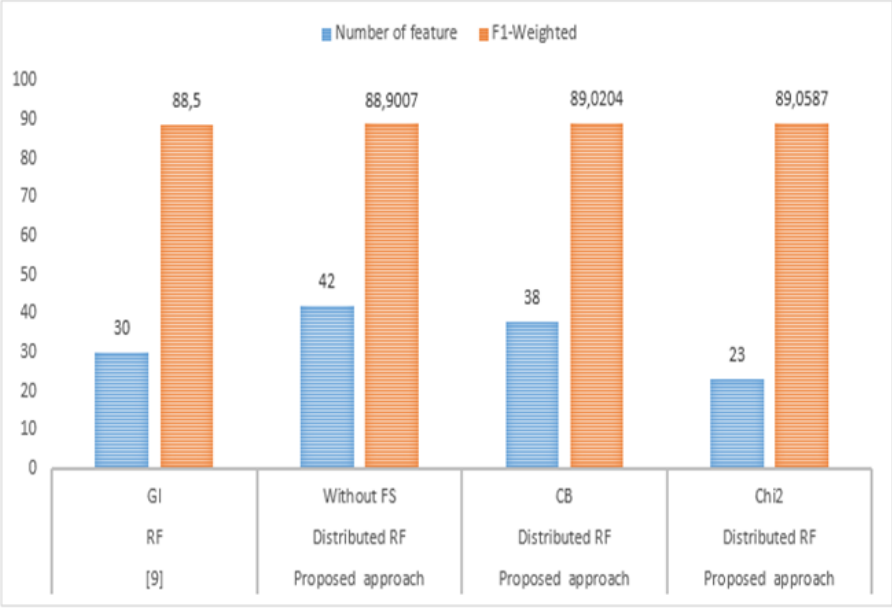

Fig. 2. Proposed approach vs [11] F1-weighted

ISSN: 2414-3103
DOI: 10.20474/japs-5.1.4

TABLE 2
F1-WEIGHTED COMPARISON BETWEEN PROPOSED APPROACH AND WORK

Ref	ML	FS	Number of Feature	Tools	F1-Weighted
[11]	RF	GI	30	Scikit-learn	88.5
Ou approachr	RF	NO	All 42	Sk-dist in Spark	88,9007
	RF	CB	38	Sk-dist in Spark	89,0204
	RF	Chi2	23	Sk-dist in Spark	89.0587

The best result obtained is when using Chi2 as a feature selection method and RF as a classification algorithm. With this combination, we have an f1-weighted score equal to 89.0587 with a reduced subset of only 23 feature. Which is much better than the result obtained by [11] (+0.5587 in f1-weighted and a reduction of 7 features in the number of features).

To ensure the efficiency of our IDS, we carried out other comparisons with other related work using the accuracy. Table 3 and Figure 3 illustrates the results of this comparison. The first four lines of the table show the results without features selection method (with all 42 features).

TABLE 3
ACCURACY COMPARISON BETWEEN PROPOSED APPROACH AND RELATED WORKS

Ref	ML	FS	Tools	Number of features	Accuracy
Proposed Approach	RF	NO	Sk-dist in Spark	42	88,974
[10]	DT		WEKA	42	87
[14]	DT		WEKA	42	81,49
[17]	DT		Visual basic	42	85,65
[10]	DT	GR	WEKA	18	88,3
[14]	DT	GALR	WEKA	20	81,42
[13]	RF	Feature significance	WEKA	5	81,618
Proposed approach	RF	Chi2	Sk-dist in Spark	23	89,143

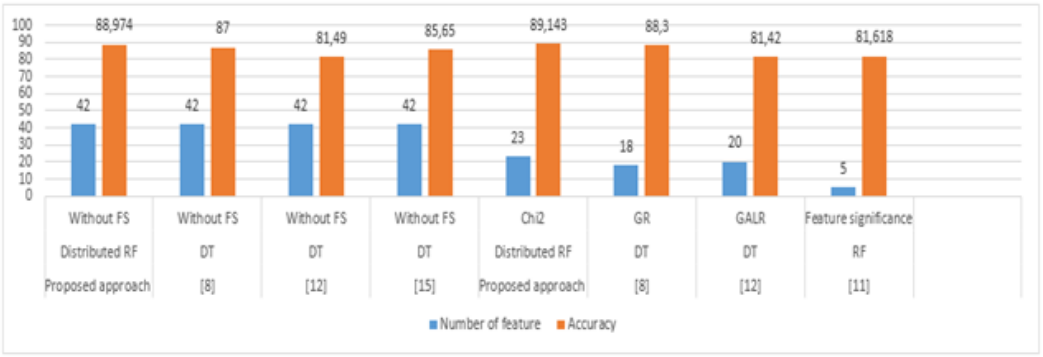

Fig. 3. Related works vs proposed approach accuracy

Our approach using the RF implementation of the Sk-dist library is much better compared to other works with an improvement which goes between 1.974 (with [10]) and arriving up to 7.487 (with [14]). The rest of the table illustrates the comparisons when using feature selection methods to reduce the original dataset. The results of the best combination obtained during the tests of our IDS (Chi2 as selector and RF as a classifier) prove their effectiveness in a matter of reduction and precision compared to the other works. Our approach arrives at an accuracy of 89.1430 using only 23 features (an improvement of 0.843 compared to [10]).

The other essential point that differentiates our approach to all the works cited is that it supports distribution in a Spark cluster which ensures high availability and fault tolerance of our IDS. The tests carried out in this point have shown that when two spark workers are used and if one of the two falls our IDS works normally with one without any interruption.

CONCLUSION

In this paper, we create a distributed and powerful intrusion detection system that allows to analyse massive data and to ensure better accuracy while using the minimum number of features during the analysis. To develop this system we

ISSN: 2414-3103
DOI: 10.20474/japs-5.1.4

combined feature selection methods (Chi2 and Correlation-based) from the Scikit-learn library, a distributed version of the random forest classifier from the Sk-dist library, a distributed package for optimizing the parameters of classifier named Hypopt and the Apache Spark framework to provide the processing cluster which is more suited to big data analysis.

For the evaluation of our approach, we opted for the UNSW-NB15 data set which better represents modern computer networks and which contains several categories of attacks. The main contribution is that our IDS overcome the major limitation of the Sk-learn library (non-distributed processing) to benefit from the multiple-choice of these algorithms in terms of classification and feature selection. From the comparison made with the other related works, we could see that our system is more efficient in terms of accuracy and speed and especially fault-tolerant (because of its distribution in a Spark cluster).

As a future work, we aim to test the machine-learning library integrated into Spark called Spark ML and compare the results with our system, also improve our IDS so that it detects the types of attack (multi-classification).

REFERENCES

[1] S. Mukkamala, A. Sung, and A. Abraham, ``Cyber security challenges: Designing efficient intrusion detection systems and antivirus tools,'' *Enhancing Computer Security with Smart Technology*, vol. 5, no. 7, pp. 125-163, 2005. doi: https://doi.org/10.1201/9780849330452.ch6

[2] H. Maizir, R. Suryanita, and H. Jingga, ``Estimation of pile bearing capacity of single driven pile in sandy soil using finite element and artificial neural network methods,'' *International Journal of Applied and Physical Sciences*, vol. 2, no. 2, pp. 45-50, 2016. doi: https://doi.org/10.20469/ijaps.2.50003-2

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, ``Scikit-learn: Machine learning in python,'' *Journal of Machine Learning Research*, vol. 12, no. 5, pp. 2825-2830, 2011.

[4] M. Chen, S. Mao, Y. Zhang, and V. C. Leung, *Big Data: Related Technologies, Challenges and Future Prospects*. Heidelberg, Germany: Springer, 2014.

[5] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis*. Newton, MA: O'Reilly Media, Inc, 2015.

[6] Apache Spark, ``Spark SQL, data frames and datasets guide,'' 2010. [Online]. Available: https://bit.ly/2Zap8tR

[7] A. H. Al-Saeedi and O. Altun, ``Binary Mean-Variance Mapping Optimization Algorithm (BMVMO),'' *Journal of Applied and Physical Sciences*, vol. 2, no. 2, pp. 42-47, 2016. doi: https://doi.org/10.20474/japs-2.2.3

[8] N. Moustafa and J. Slay, ``UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),'' in *Military Communications and Information Systems Conference (MilCIS),* California, CA, 2015.

[9] Git Hub, ``Sk-dist: Distributed scikit-learn meta-estimators in Py Spark,'' 2019. [Online]. Available: https://bit.ly/2Clb9bp

[10] H. M. Anwer, M. Farouk, and A. Abdel-Hamid, ``A framework for efficient network anomaly intrusion detection with features selection,'' in *9th International Conference on Information and Communication Systems (ICICS),* London, UK, 2018.

[11] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, ``Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives,'' in *IEEE 3rd International Conference on Computing, Communication and Security (ICCCS),* Bangkok, Thailand, 2018.

[12] B. W. Yap, K. Abd Rani, H. A. Abd Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, ``An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets,'' in *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013),* Berlin, Germany, 2014.

[13] T. Janarthanan and S. Zargari, ``Feature selection in UNSW-NB15 and KDDCUP'99 datasets,'' in *IEEE 26th International Symposium on Industrial Electronics (ISIE),* New York, NY, 2017.

[14] C. Khammassi and S. Krichen, ``A GA-LR wrapper approach for feature selection in network intrusion detection,'' *Computers & Security*, vol. 70, pp. 255-277, 2017. doi: https://doi.org/10.1016/j.cose.2017.06.005

[15] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, ``A novel two-stage deep learning model for efficient network intrusion detection,'' *IEEE Journal*, vol. 7, pp. 30 373-30 385, 2019. doi: https://doi.org/10.1109/access.2019.2899721

ISSN: 2414-3103
DOI: 10.20474/japs-5.1.4

[16] N. Moustafa and J. Slay, ``A hybrid feature selection for network intrusion detection systems: Central points,'' in *Proceedings of the 16th Australian Information Warfare Conference,* Sydney, Austraila, 2017.

[17] N. Moustafa and J. Slay, ``The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set,'' *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18-31, 2016. doi: https://doi.org/10.1080/19393555.2015.1125974

[18] Scikit Learn, ``Sklearn feature selection chi2 scikit-learn 0.22 documentation,'' 2019. [Online]. Available: https://bit.ly/2Z86ene

ISSN: 2414-3103
DOI: 10.20474/japs-5.1.4