Chapter

Intrusion Detection Based on Big Data Fuzzy Analytics

Farah Jemili and Hajer Bouras

Abstract

In today's world, Intrusion Detection System (IDS) is one of the significant tools used to the improvement of network security, by detecting attacks or abnormal data accesses. Most of existing IDS have many disadvantages such as high false alarm rates and low detection rates. For the IDS, dealing with distributed and massive data constitutes a challenge. Besides, dealing with imprecise data is another challenge. This paper proposes an Intrusion Detection System based on big data fuzzy analytics; Fuzzy C-Means (FCM) method is used to cluster and classify the pre-processed training dataset. The CTU-13 and the UNSW-NB15 are used as distributed and massive datasets to prove the feasibility of the method. The proposed system shows high performance in terms of accuracy, precision, detection rates, and false alarms.

Keywords: Intrusion detection, machine learning, Apache Spark, Big Data, CTU-13, UNSW-NB15, Feature selection, FCM clustering

1. Introduction

Recently, in computer networks the numbers of intrusions have grown extensively, and many new pirating tools and intrusive methods have appeared. To save the security of computer systems, several solutions have been identified like intrusion detection systems (IDS) which it is the mean solution to deal with suspicious activities in a network [1].

Using IDS tools, the presence of imperfect information greatly influences the response data under non-suitable as a medium for decision-making. Uncertainty is presented as imperfect data, the variability of the data that resides in the random nature of the information due to the heterogeneity of data sources, vagueness and incompleteness of data due to the lack of useful data [2]. Thus, fuzzy clustering as a robust artificial intelligent method has been successfully employed to reduce the amount of false alarm generated by the detection process and separate the overlap between normal and abnormal behavior in computer networks [3].

Hence, we use two intrusion detection datasets CTU-13 and UNSW-NB15 which contain varieties of intrusions, that we combine into one homogenous dataset and then we apply our ML model based on the Fuzzy C-Mean (FCM) clustering algorithm. We choose Microsoft Azure Blob Storage to load our datasets on.

This paper addresses the problem of generating application clusters from the network intrusion detection datasets. The Fuzzy C-Mean (FCM) clustering algorithms were chosen to be used in building an efficient network intrusion detection model. The paper is structured as follows: Section 2 provides related work of IDS using Big Data techniques, Section 3 introduces brief introduction about intrusion detection, Section 4 presents the used datasets, the proposed system and its components, Section 5 illustrates the evaluation metrics and results of the tested system, finally, Section 6 provides conclusions and further development of future work.

2. Related work

Several works of IDS using Big Data techniques exist. Jeong et al. [4] indicate that Hadoop can solve intrusion detection and big data issues by focusing specifically on anomalous IDSs. The experience of Lee et al. [5] with Hadoop technologies shows good feasibility as an intrusion detection instrument because they were able to reach up to 14 Gbps for a DDOS detector. M. Essid and F. Jemili [6] have combined and eliminated the redundancy of the alerts bases KDD99 and DARPA, they used Hadoop for data fusion. Besides, R. Fekih and F. Jemili [7] used Spark to merge and remove the redundancy of the three alerts bases KDD99, DARPA and MAWILAB. The main objective was to improve detection rates and decrease false negatives. Terzi et al. [8] created a new approach to unsupervised anomaly detection and used it with Apache Spark on Microsoft Azure (HDInsight21) to harness scalable processing power. The new approach was tested on CTU-13, a botnet traffic dataset, and achieved an accuracy rate of 96%. M.Hafsa and F.Jemili [9] created a new approach to intrusion detection. They used Apache Spark on Microsoft Azure (HDInsight21) to analyze and process data from the MAWILAB database. Their new approach achieved an accuracy rate of 99%. Ren et al. [10] created a new approach to unsupervised anomaly detection using the KDD'99 base to analyze and process the data, they achieved a low detection rate. Rustam and Zahras [11] compared two models, one supervised (the Support Vector Machine SVM model) and the other unsupervised (Fuzzy C-Means FCM) to analyze, process, and detect KDD'99 database intrusions. They found that SVM achieved an average accuracy rate of 94.43%, while FCM achieved an average accuracy rate of 95.09%. In this work, we propose an Apache Spark-based approach to detect intrusions. The goal of our system is to provide an efficient intrusion detection system using Big Data tools and fuzzy inference to treat uncertainties and provide better results.

3. Intrusion detection system (IDS)

Intrusion: An intrusion is any use of a computer system for purposes other than those intended, usually due to the acquisition of privileges illegitimately. The intruder is generally seen as a stranger to the computer system that has managed to gain control of it, but statistics show that the most common abuses come from internal people who already have access to the system [12].

Intrusion detection: Intrusion detection has always been a major concern in scientific papers [13, 14]. By security auditing mechanisms, it consists in analyzing the collected information in search of possible attacks.

Intrusion detection system (IDS): To detect and signal anomalous activities, an intrusion detection system (IDS) is used. To detect and signal anomalous activities, an intrusion detection system (IDS) is used. This system protects a system from malicious activities coming from known or unknown sources, this process is done automatically in order to protect confidentiality, integrity and availability of systems. Cannady et al. [15] states that an IDS has two detection approaches: an anomaly-based detection and signature-based detection. IDS are characterized by their surveillance domain which can monitor a corporate network, multiple machines or applications.

4. Data and tools

4.1 Data sets

4.1.1 CTU-13 data set

CTU-13 consists in a group of thirteen scenarios that each run a specific botnet performed in a real network environment. Each scenario includes a botnet pcap file, a tagged NetFlow file, a README file with the capture timeline, and the malware run file. The NetFlow (network flow) file is based on bidirectional flows that provide information about the communication between a source (a client) and a destination (a server). This dataset includes three types of traffic with a different distribution: Normal, botnet (or Malware), and background:

- **Normal traffic:** comes from normal hosts that are previously verified and very useful to check the actual performance of machine learning algorithms.
- **Malware or botnet traffic:** comes from malicious hosts or robots.
- **Background traffic:** is considered as unknown traffic that is needed to saturate the algorithms in order to check their speed performance and test if the algorithm merges with the other traffic [16].

The following **Figure 1** presents the distribution of experimental data for CTU-13 data set:

4.1.2 UNSW-NB15 data set

UNSW-NB 15 is a dataset that was created in an Australian Cyber Range Lab using an IXIA PerfectStorm tool to extract a hybrid of realistic modern natural activities and contemporary synthetic attack behaviors generated by network traffic. This dataset contains 49 features are categorized into five groups and which are explained in [17, 18].

The following **Table 1** represents the attack types which are classified into nine groups.

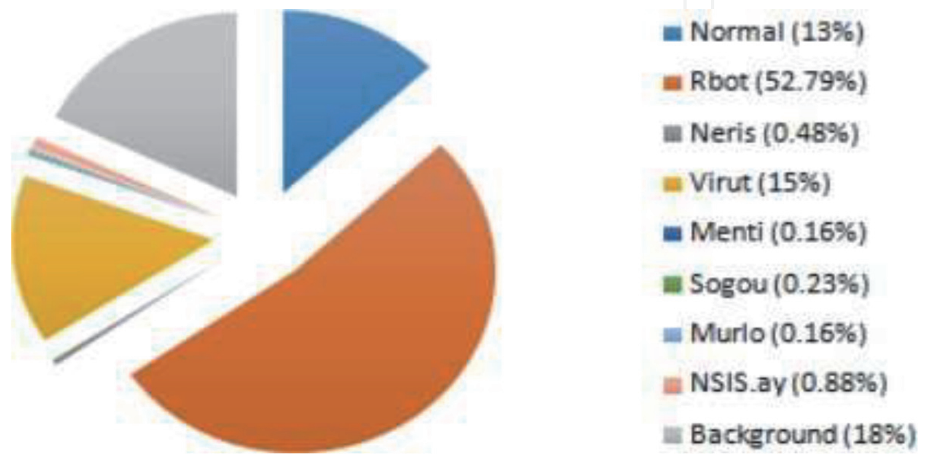


Figure 1.
Attack categories in CTU-13.

Types	Description
Fuzzers	The attacker attempts to cause a program or network suspended by feeding it the randomly generated data.
Analysis	It penetrates the web applications via ports (port scan), web scripts (HTML files), and emails (spam).
Backdoor	A technique in that a system security mechanism is bypassed to access a computer or its data.
DoS	An intrusion which attempts to make a network resource or a server unavailable to users, generally by temporarily suspending the services of a host connected to the Internet.
Exploit	It takes advantage of a glitch, bug, or vulnerability to be caused by an unintentional behavior on a network or a host.
Generic	A technique establishes against every block-cipher using a hash function to collision without configuration of the block-cipher.
Reconnaissance	It gathers information about a computer network to evade its security controls.
Shellcode	The attacker penetrates a slight piece of code starting from a shell to check the compromised machine.
Worm	The attacker replicates itself in order to advance on other computers. Frequently, it uses a computer network to spread itself, relying on the security failures on the target computer to access it.

Table 1.
UNSW-NB15 attack types.

4.2 Apache spark

Apache Spark, is powerful hybrid, scalable and fast distributed data processing engine most active open source project in big data. It was developed at UC Berkeley in 2009. It became one of the top projects in Apache in 2010 [19]. Spark provides APIs in Scala, Java, Python and R languages. To get a good hold on huge data, it must be fast enough by processing massive data at once. Therefore, it is necessary that Spark is available on several clusters rather than on a single machine. The result of the treatment provided by Spark is not written to the disk but kept in memory. This all-in-memory ability is a high-performance computing technique for advanced analytics, making Spark 100 times faster than Hadoop (**Figure 2**) [20].

Spark also has an ecosystem of libraries that can be used for Machine Learning, interactive queries. Which can have important implications for productivity. The project has been progressively enriched to provide a complete ecosystem today which is shown in **Figure 3**.

4.3 Microsoft azure

Microsoft Azure, formally known as Windows Azure, is a cloud computing platform for building, deploying and managing services and applications anywhere with the help of a global network of managed data centers located in 54 regions around the world [21]. Microsoft's HDInsight is a managed Hadoop service in Azure Cloud that uses the Hortonworks Data Platform (HDP). HDInsight clusters can be customized easily by adding additional packages and can scale up in case of high demand by allocating more processing power [22]. By the Azure Active Directory, The data is protected and persists even after the cluster is deleted.

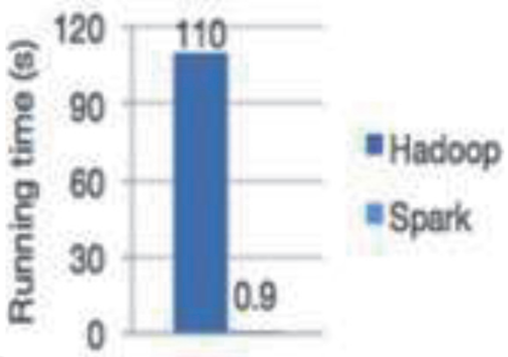


Figure 2.
Speed comparison chart between spark Hadoop.

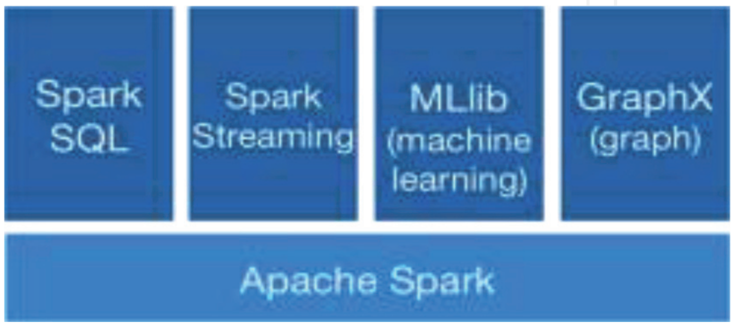


Figure 3.
Apache spark ecosystem.

4.4 Fuzzy C-means clustering (FCM)

The FCM algorithm is one of the most widely used fuzzy clustering algorithms [23] which attempts to partition a finite collection of elements into a collection of c fuzzy clusters with respect to some given criterion. This algorithm is based on minimization of the following objective function:

$$Jm = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\|x_i - c_j\right\|^2, 1 \le m < \infty \tag{1}$$

where:

- m: Any real number greater than 1.
- u_{ij}: The degree of membership of xi in the cluster j.
- x_i: The ith of d-dimensional measured data.
- c_j: The d-dimension center of the cluster.
- ||*||: Any norm expressing the similarity between any measured data and the center. The algorithm FCM is composed of the following steps:

Step1: U0, Initialize U = [uij] matrix.
Step2: At k-step, calculate the centers vectors C (k) = [cj] with U(k) [24]

$$cj = \frac{\sum_{i=1}^{N} xiu_{ij}^{m}}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{2}$$

```
The FCM Algorithm
  begin
  Fix $c, 2 < c < n$;
  Fix $\epsilon$, (e.g., $\epsilon = 0.001$);
  Fix maxIerations, (e.g., maxIerations=100);
  Choose any inner product norm metric (e.g., Euclidean distance);
  Fix $m, 1 < m < \infty$, (e.g., $m = 2$);
  Randomly initialize $V_0 = v_1, v_2, \ldots, v_c$ cluster centers;
  for $t = 1$ to maxIterations do
     Update the membership matrix $U$ using Eq. 3;
     Calculate the new cluster centers $V^t$ using Eq. 2;
     Calculate the new objective fucntion $J_m^t$ using Eq. 1;
  end for
  end
```

Figure 4.
Pseudo code of FCM algorithm-.

Step3: Update U (k), U(k + 1).

$$cj = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|xi-cj\|}{\|xi-ck\|} \right)^{\frac{2}{m-1}}} \tag{3}$$

Step4: If || U(k + 1) -U(k) || < *s* then STOP, else return step 2.

Step5: The Fuzzy partitioning [25] is realized out through an iterative optimization of the objective function in Eq. (1), with the cluster centers cj by using Eqs. (2) and (3) and the update of membership uij.

Step6: This iteration will stop when:

$$max_{\,ij} \left\| u_{ij}^{k+1} - u_{ij}^{k} \right\| < \epsilon \tag{4}$$

Where:

- ε: Termination criterion between 0 and 1.
- k: The iteration steps.

A pseudo code of the algorithm FCM is presented as follows (**Figure 4**).

5. Proposed method

The idea of our distributed architecture comes down to a process adaptation of in data fusion approach. This architecture allows us to facilitate data analysis with a powerful Spark big data tool (see **Figure 5**).

5.1 Converting incoming file

We will be using Jupyter Notebook with Apache Spark and the Python API (PySpark). In this stage, we will read CSV files and converting them to Apache Parquet format into Microsoft Azure Blob Storage. Apache Spark supports multiple operations on data, it bids the ability to convert data to another format in just one line of code. Developed by Twitter and Cloudera, Apache Parquet is an open-source columnar file format optimized for query performance and minimizing I/O, offering very efficient

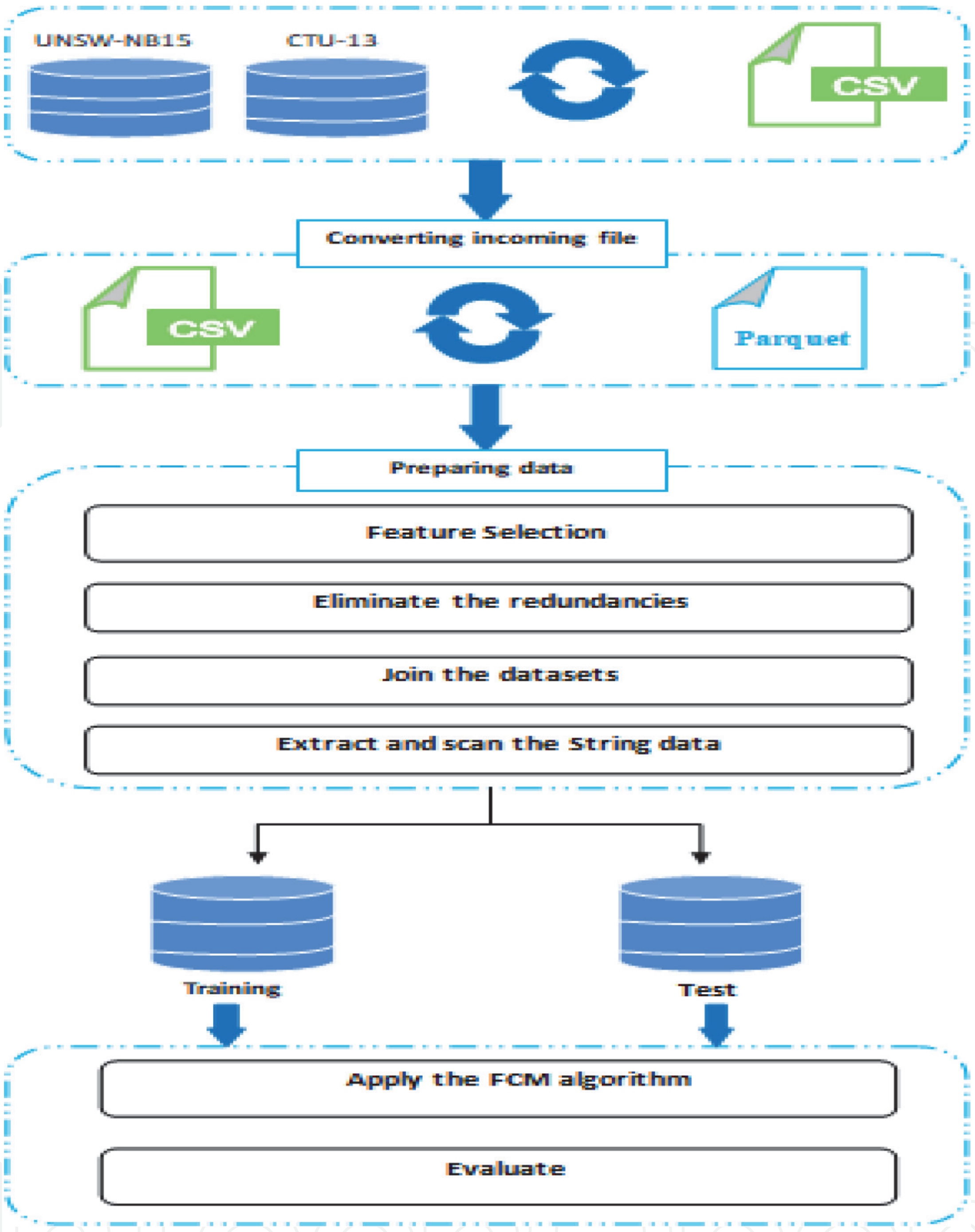


Figure 5.
Diagram of proposed approach.

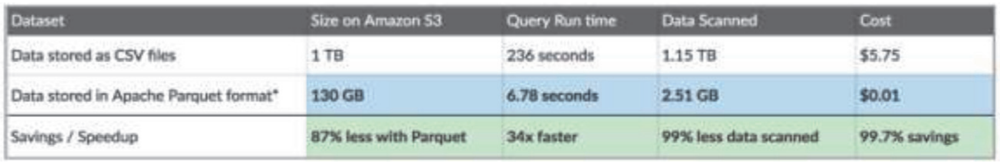
compression and encoding schemes [26]. **Figure 6** shows the efficiency of using the Parket format. This format minimizes storage costs and data processing time.

The following **Table 2** indicates the old and new size of each datasets after converting to Apache Parquet, We notice that by converting CSV to Parquet the storage costs are minimized.

5.2 Preparing data

5.2.1 Feature selection

The feature selection phase selects relevant attributes required for decision making. A pre-processing phase converts the flow records in a specific format which is acceptable to an anomaly detection algorithm [27].



Dataset	Size on Amazon S3	Query Run time	Data Scanned	Cost
Data stored as CSV files	1 TB	236 seconds	1.15 TB	$5.75
Data stored in Apache Parquet format*	130 GB	6.78 seconds	2.51 GB	$0.01
Savings / Speedup	87% less with Parquet	34x faster	99% less data scanned	99.7% savings

Figure 6.
Apache parquet advantages.

DataSets	Average Size (CSV)	Average Size (Parquet)	speedup
CTU-13	2600.96MO	555MO	x4.69
UNSW-NB15	559MO	202MO	x2.77

Table 2.
Average file size before and after converting.

with the CTU-13 dataset, We did not utilize the feature selection algorithm for this dataset, we instead selected columns that were pertinent and delete unnecessary features(empty columns). After the removing, we get with a total of 13 columns.

Using UNSW-NB15 dataset, we processed the data selection problem. We apply a combination fusion of Random Forest Algorithm with Decision Tree Classifier. V. Kanimozhi [28] decides that the combined fusion of these two algorithms provides 98.3% has listed the best four features are as sbytes, sttl, sload, ct_dst_src_ltm and the **Figure 7** labels the graphical representation of Feature Importances and the top four features.

The goal of eliminating no-useful attributes is bring about a better performance by the system with a better accuracy.

5.2.2 Eliminate the redundancies

This eliminate redundancies task involves removing duplicates (removing all the repeated records) which helps with attack detection as it makes the system less biased by the existence of more frequent records.This tactic makes computation faster as it must deal with less data [29].

5.2.3 Join the datasets

Before the merge of the bases, some common columns have different names from one database to another (for example “Label” in CTU-13 named “attack_cat” in UNSW-NB15), in this state, we will rename these attributes then we will merge our bases. Since, Apache Spark offers the ability to join our databases in just one line of code. The following Listing shows the query used:

Listing 1: Merge databases

```
FinalData = CTU_13.join(UNSW_15,on=['dur','proto','sport'
 'dsport','state','sbytes','Label'],how='full_outer')
```

5.2.4 Extracting and scanning string data

Using the Apache Spark Machine Learning library, we create a Machine Learning pipeline. A pipeline is a sequence of stages where each stage is either an Estimator or a Transformer.

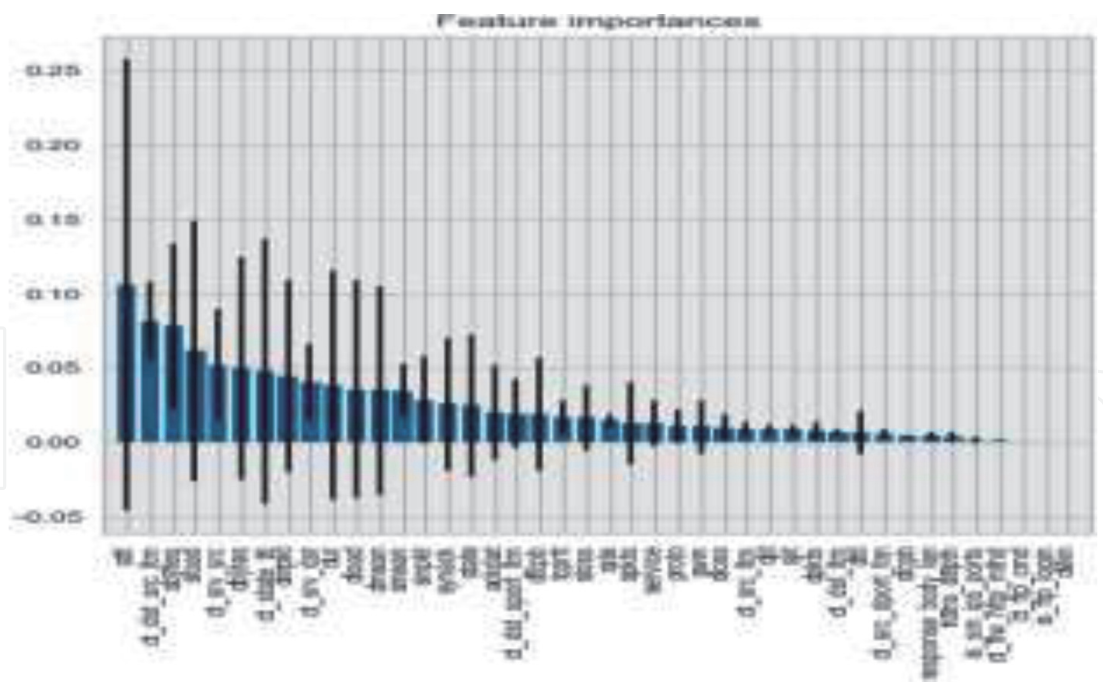


Figure 7.
Feature importance of UNSW-NB15 dataset.

In our final base, some attributes are of types string (Like: Label, sport, proto, ...), in this step we will convert all attributes of type string to attribute of type integer by using the transformer "StringIndexer" which encodes a string column of labels to a column of label indices. These indices are ordered by label frequencies, the most frequent label gets index 0.

StringIndexer classifies attacks automatically in class, it assigns the same index for attacks of the same category.

5.3 FCM application

In our experimental work and as we said above we will use Microsoft Azure as a cloud environment to upload and analyze the dataset with FCM algorithm. We use the training dataset to form and evaluate our model. The test dataset is then used to make predictions. We choose to train our Model with FCM algorithm. The first stage of the FCM algorithm is to initialize the input variable, the input vector includes the dataset features, the number of cluster is 2 (**1 = intrusion and 0 = normal**), and the center of cluster is calculated by taking the means of all feature in the final dataset.The use of fuzzy C-means clustering algorithm to classify data will generate a number of clusters, each cluster contains part of the data records [30]. The characteristics are different between normal and intrusion data records, so they should be in different clusters as shown in the following **Figure 8** which presents the data records clustering.

5.4 Performance metrics

Apache Spark Machine Learning provides a suite of metrics to evaluate the performance of Machine Learn- ing models [31]. To measure the performance in our work the metrics used are as present in below **Table 3** (Where TP = True Positives, TN = True Negatives, FP = False Positives and FN = False Negatives).

After apply the FCM to our final dataset(After merging our intrusion detection datasets) the result is shown in the following **Table 4**.

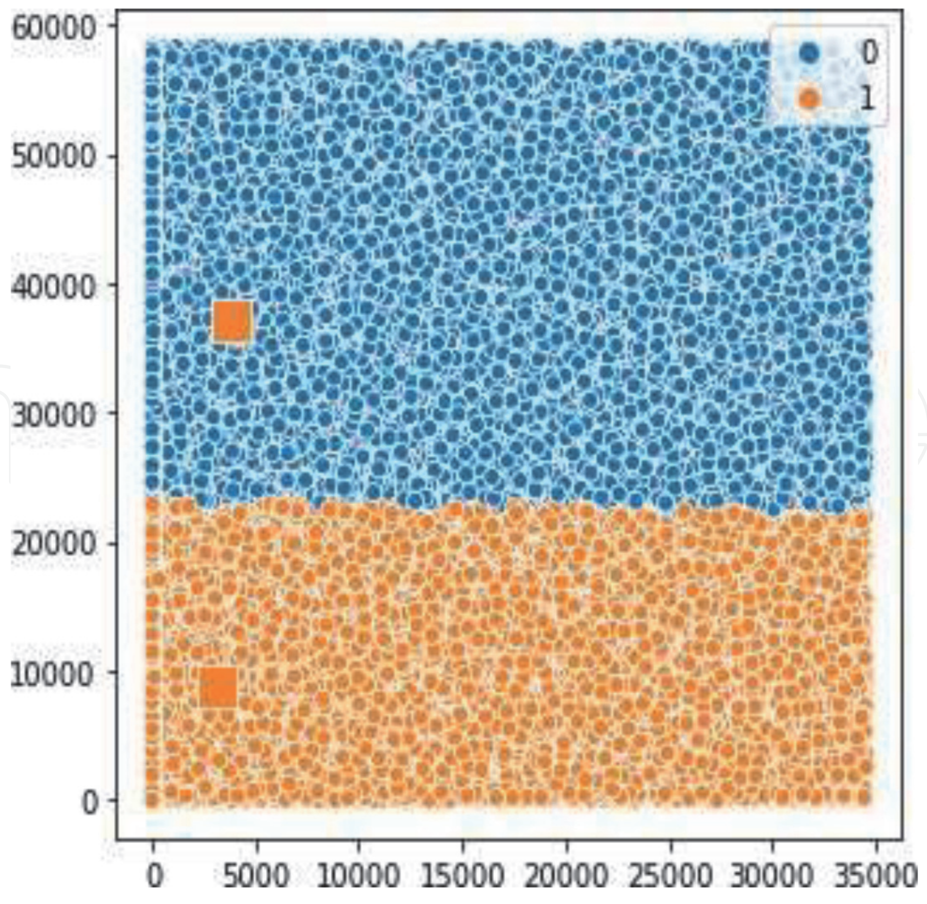


Figure 8.
The data records clustering.

Measure	Description	Formula
Accuracy	Accuracy measures performance across all labels	Accuracy = TP + TN/ TP + FP + FN + TN
Precision	The ratio of correctly predicted positive observations to the total predicted positive observations.	Precision = TP/TP + FP
Recall	The ratio of correctly predicted positive observations to the all observations in actual class	Recall = TP/TP + FN
F-measure	The weighted average of Precision and Recall	F1 = 2*(Recall * Precision) / (Recall + Precision)

Table 3.
Evaluation metrics.

As shown in **Table 4** the total input data is 845 721 records, 243 899 records as normal and 601 822 records as intrusion. After applying FCM algorithm, the result is 231 704 record for normal and 589 785 records for intrusion. Then we calculated the normal and intrusion classification rate by the following equation:

$$Classification\ rate = \frac{Number\ of\ classified\ patterns}{Total\ number\ of\ patterns} * 100 \tag{5}$$

The simulation results show that the classification rate is 96.4% by the FCM algorithm which means that the false positive rate(returns the rate of instances which are falsely classified) is 0.02%.

	Input data	Output data	Classification rate
Normal	243 899	231 704	94.9%
Intrusion	601 822	589 785	97.9%

Table 4.
Evaluation metrics.

6. Discussion

It is possible to obtain a very precise system (accuracy of 99%) but not very efficient with a recall of 10%. In our work, with an accuracy of 97.2% and a recall of 96.4%, we can say that our system is efficient. The use of the fuzzy algorithm in this experiment gave a good result. The advantage of our system is the fuzzy representation that is increasingly used to deal with missing and inaccurate data problems which is the disadvantage of most classification algorithms.

7. Conclusions and future work

In this paper, we achieved a successful distributed IDS. Using the FCM algorithm allows to effectively train and analyze our model after merging datasets. Proposing a distributed system and showing the power of Spark to combine and handle large and heterogeneous structures of training datasets present the main merits in our work.

In future work, we will perform our dataset analysis with another Big Data framework expected to reach faster results. In addition, we will develop our approach with other classifiers to get better results.

Author details

Farah Jemili[1]* and Hajer Bouras[2]

1 ISITCom, Mars Research Laboratory, University of Sousse, Hammam Sousse, Tunisia

2 ISITCom, University of Sousse, Hammam Sousse, Tunisia

*Address all correspondence to: jmili_farah@yahoo.fr

References

[1] Hafsa, M., Jemili, F. (2018). "Comparative Study between Big Data Analysis Techniques in Intrusion Detection. Big Data and Cognitive Computing", 2018.

[2] https://reference.wolfram.com/legacy/applications/fuzzylogic/Manual/12.html.

[3] D. Song, M.I. Heywood, A.N. Zincir-Heywood, "Training Genetic Programming on Half a Million Patterns: An Example from Anomaly Detection," IEEE Transactions on Evolutionary Computation, 2005.

[4] H. W. L. J. Y. I. Jeong H, «Anomaly teletraffic intrusion detection systems on hadoopbased platforms: A survey of some problems and solutions,» 15th international conference on. IEEE, pp. 766-770.

[5] L. Y. Lee Y, «Toward scalable internet traffic measurement and analysis with hadoop,» ACM SIGCOMM Comput Commun Rev, vol. 43(1), pp. 5-13.

[6] F. J. Mondher Essid, «Combining intrusion detection datasets using MapReduce,» In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 10/2016.

[7] Ben Fekih, R., & Jemili, F. Distributed Architecture of an Intrusion Detection System Based on Cloud Computing and Big Data Techniques, 2018.

[8] D. S. Terzi, R. Terzi and S. Sagiroglu, "Big data analytics for network anomaly detection from netflow data," in International Conference on Computer Science and Engineering (UBMK), Antalya, 2017.

[9] Hafsa, M., Jemili, F. (2018). "Comparative Study between Big Data Analysis Techniques in Intrusion Detection. Big Data and Cognitive Computing", 2018.

[10] Ren, W., Cao, J., Wu, X. (2009). "Application of network intrusion detection based on Fuzzy C-means clustering algorithm", 3rd International Symposium on Intelligent Information Technology Application, IITA 2009.

[11] Rustam, Z., & Ariantari, N. P. A. A. (2018). Comparison between support vector machine and fuzzy Kernel C-Means as classifiers for intrusion detection system using chi-square feature selection, 2018.

[12] Moustafa, N., Slay, J. (2015). "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). 2015 Military Communications and Information Systems Conference", 2015.

[13] Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino. "An empirical comparison of botnet detection methods", 2014.

[14] Moustafa, N., Slay, J. (2016). "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", 2016.

[15] Dataricks. About Databricks. Available online: https://databricks.com/spark/about (accessed on 6 May 2018).

[16] Gaied, I., Jemili, F., Korbaa, O. (2016). "Intrusion detection based on Neuro-Fuzzy classification. Proceedings of IEEEACS International Conference on Computer Systems and Applications", AICCSA, 2016.

[17] Massimiliano, A.; Erbacher, R.F.; Jajodia, S.; Persia, M.C.F.; Picariello, A.; Sperli, G.; Subrahmanian, S.V.

Recognizing unexplained behavior in network traffic. Netw. Sci. Cybersecur. 2013.

[18] B. C.* Rhodes, J. A. Mahaffey and D. J. Cannady, “Multiple Self-Organizing Maps for Intrusion Detection," in National Information Systems Security Conference, Baltimore, 2000.

[19] Microsoft. Azure Regions. Available online: https://azure.microsoft.com/en-us/global-infrastructure/ regions/ (accessed on 5 May 2018).

[20] Neenu Daniel & Ritty Jacob, Intrusion Detection Techniques in Big Data: A Review, April 2017.

[21] Ren, W., Cao, J., Wu, X. (2009). “Application of network intrusion detection based on Fuzzy C-means clustering algorithm”, 3rd International Symposium on Intelligent Information Technology Application, IITA 2009.

[22] Mllib Evaluation Metrics. Available online: https://spark.apache.org/docs/ 2.1.0/mllib-evaluation-metrics. html (accessed on 3 June 2018).

[23] Jawhar, M. M. T., Mehrotra, M. (2010)." Design Network Intrusion Detection System using hybrid Fuzzy-Neural Network. International Journal of Computer Science and Security”, 2010.

[24] P. Biondi, «Architecture expérimentale pour la détection d’intrusions dans un système informatique», 2001.

[25] Ar, L.; Levent, E.; Vipin, K.; Aysel, O.; Jaideep, S. A comparative study of anomaly detection schemes in network intrusion detection. In Proceedings of the SIAM Conference on Applications of Dynamical. Systems, 2003.

[26] Premasundari, M., Yamini, C. (2019). a Violent Crime Analysis Using Fuzzy C-Means Clustering Approach, 6956(April), 2019.

[27] K. Taha and P.D. Yoo, “SIIMCO: A Forensic Investigation Tool for Identifying the Influential Members of a Criminal Organization”, IEEE Transactions on Information Forensics and Security, 2016.

[28] Apache Parquet vs. CSV Files—DZone Database. Available online: https://dzone.com/articles/how-to-bea-hero- with-powerful-parquet-google-and (accessed on 6 February 2018).

[29] Umer, M. F., Sher, M., Bi, Y. (2017). “Flow-based intrusion detection: Techniques and challenges. Computers and Security”, 2017.

[30] Kanimozhi, V., Jacob, P. (2019). “UNSW-NB15 dataset feature selection and network intrusion detection using deep learning. International Journal of Recent Technology and Engineering”, 2019.

[31] Verma, R.; Kantarcioglu, M.; Marchette, D.; Leiss, E.; Solorio, “Security Analytics: Essential Data Analytics Knowledge for Cybersecurity Professionals and Students”, IEEE Secur. Priv. 2015.