International Journal of Engineering Sciences & Emerging Technologies, Sept. 2017.
ISSN: 22316604 Volume 9, Issue 6, pp: 225-234 ©IJESET

# INTEGRATED MODEL FOR HANDLING ABNORMAL NETWORK CONNECTIONS USING PARALLELIZING K-MEANS (PKM) CLUSTERING AND BIG DATA PROCESSING TOOL (SPARK)

Ahmed Fuad Al-Dubai[1], Vikas T. Humbe [2], Santosh S. Chowhan [3]
[1]Department of Computational Sciences and Technology,
S.R.T.M University, Nanded City, India Country
Ahmed_Aldubai86@yahoo.com
[2] School of Technology, S.R.T.M University, latur City, India Country.
vikashumbe@gmail.com
[3] Department of Computational Sciences and Technology,
S.R.T.M University, Nanded City, India Country.
drschowhan@gmail.com

***ABSTRACT***

*Due to speedily growth technologies of network communication, the internet has emphasized the need to guarantee the security of sensitive and real-time data passing through networks. Several cryptography methods are considered to build the security mechanism of Big data, but is not sufficient due to the increments of the hacker's computation ability those cryptography methods are going to be broken down in the nearly future. This paper proposed a novel integrated model lies in use of Parallelizing k-means (PKM) clustering and Apache Spark, a Big data processing tool with reduced feature technique, that can handle abnormal network connections and misplaced data packages which can reduce the overall accuracy of classification and increase the Prediction Time. Five machine learning algorithms are used in the comparison. The proposed model provides superior performance as compared to other two models.*

***KEYWORDS:*** *Big Data, Intrusion Detection, Machine Learning Algorithms, Parallelizing k-means, & Spark*

## I. INTRODUCTION

With the advent of digital technology, the size of the data being generated every second has been crossing the boundary of gigabytes and even into terabytes. Companies from different domains are gaining profit by managing their resources and transactions over the network. Thus, Big data security remains a key challenge for all the solutions because data value is no worth if we compromise with data security and privacy. One of most important security challenging issues over real-time data is to detect and prevent network intrusions with less prediction time since the network is the backbone of Big data. These intrusions affect the confidentiality, integrity, and availability of Big data resources and offered services. some providers of Big data services use the firewall to find a solution for above issues. Firewall is treated as the first line of defense, can only sniff the packets at the border of a network (outsider attacks), insider attacks cannot be detected. several attacks are too complex to detect using a traditional firewall [1]. Thus, the traditional firewall is not an efficient solution to block all the intrusions. To define an efficient solution for such problems, a high-speed intrusion detection system should be capable of work in Big data environment and process huge data of network traffic at the same time, since it acts as an additional preventive layer of security.

## II. RELATED WORK

Hacking in the form of a cyber attack has appeared intimately in the news. Some attacks take over network traffic and push out normal traffic. However, there are attacks that exploit unauthorized privileges to the computer, such as the flaws of networking software. At this time, it is very difficult to know whether the attacked computer is attacked. This is because it is necessary to find abnormal attacks between so many network requests in order to find such exploits. Some attacks follow certain known patterns. For example, accessing all possible ports in a short period of time is a pattern that generic software does not normally do. However, this is generally the first step that attackers who are looking for a computer to do exploit are usually the first step.

If a connection attempt is made to several ports over a short period of time, a few connection attempts can be regarded as common, but most attempts will be abnormal and we can judge this as a port-scanning attack. These known types of attacks can be detected and detected in advance. But what happens if an unknown type of attack comes in? The biggest problem is that it may be some form of attack. In the meantime, other forms of access attempts should be considered as potential attack traffic and monitored [2].

Unsupervised learning can be used here with help of PKM clustering and Spark tool. Using previous methods and tools, abnormal network connections can be detected easily. When a network connection is clustered using PKM clustering and a different connection is requested from the existing normal network connection cluster, this can be regarded as an abnormal connection.

Spark has emerged as an important platform in the area of Big data for data-intensive computing of huge datasets. It is an open-source platform, developed by UC Berkeley AMP Lab and licensed by Apache for data storage and processing [3]. Tan Z et al [4]. Design a collaborative intrusion detection scheme for enhancing Big data security. Gunasekaran, R., et al [5]. Implement a method to analyze and correlate the different type of login real-time. Solaimani, M., L. Khan, et al [5] proposed framework for a real-time anomaly detection using Apache Storm. There are a few number of researchers studies which focus on network intrusion detection over fast streaming data using Hadoop ecosystem. Prathibha and Dileesh [7] Designed hybrid intrusion detection system using Snort rules on Hadoop. However, they did not provide complete implementation details and the accuracy results. likewise, Bandre and Nandimath [8] design consideration of Network Intrusion Detection System based on the Hadoop framework. They used seven parameters for IDS and, General Purpose Graphical Processing Unit (GPGPU) to accelerate the performance. Their work considers three attacks i.e., Arpaio, Denial of Service (DoS), and Tcp_Syn. At the end, they did not provide the accuracy of the system. Jeong et al. [8] Discuss different techniques proposed in the literature and few works done by Hadoop systems but, most of them are still not efficient enough to process high-speed Big data at a real-time. Therefore, based on the previous ML knowledge, our proposed model can detect intrusions using 13 selected features with higher accuracy and less period of time.

## III. AN INTEGRATED MODEL FOR HANDLING ABNORMAL NETWORK CONNECTIONS

The main purpose of our integrated model is to take help of data mining techniques such as feature selection, eliminating redundancies and normalization in order to analyze the huge data like the KDDCUP'99. The second purpose is to reduce the prediction time to handle abnormal network connections by using PKM clustering and an open-source Apache Spark tool designed for a Big data processing and the third one is the classification algorithms that realizes knowledge wave characters. This integrated model is distinguished by its powerfulness thanks to its huge database and rapidly prediction due to its parallel architecture. In addition, to test out the adaptability of our model in handle abnormal network connections, we use 10-fold cross-validation test, that work by breaking dataset into ten sets of size n/10, nine sets used as a training and one set used as testing, this process is repeated ten times after that a mean accuracy is taken. lastly, we prove that our integrated model based on PKM clustering and spark is more powerful in classifying intrusions and prediction time than others models.

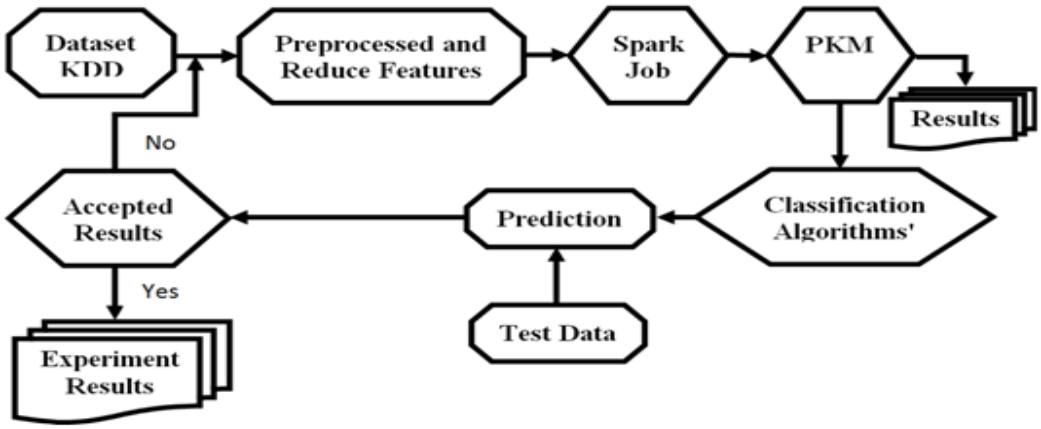


**Figure 1.** Our Integrated Model for Handling Abnormal Network Connections

### 3.1. Data Set Description

The kddcup99 dataset was used in the experimental evaluation. It contains five main categories and 20 subcategories of attacks [10]. The total number of instances included in training and testing dataset are 500000 instances. Which fulfill the Big data criteria of volume for Intrusion Detection. Several researchers applied KDDcup99 for the comparison of Machine Learning Algorithms.

### 3.2 Information Gain For Features Selection

Information gain evaluates the worth of an attribute by measuring the information gain with respect to the class.

$$InfoGain(Class,Attribute) = H(Class) - H(Class \mid Attribute) \quad (1)$$

Which can calculate the information gain for each attribute for the output variable. The values of entry can be 0 (no information) or 1(maximum information). Attribute that gives more information will have a higher information gain value and can be selected, whereas those that do not add much information will have a lower score and can be removed [12]. We have applied this algorithm over our dataset and the features are reduced from 41 to 13 features only.

**Table 1:** A list of Feature Selection by IG Algorithm

| Attributes: 13 |
|---|
| **protocol_type** |
| **service** |
| **flag** |
| **src_bytes** |
| **land** |
| **wrong_fragment** |
| **hot** |
| **logged_in** |
| **count** |
| **serror_rate** |
| **diff_srv_rate** |
| **dst_host_same_src_port_rate** |
| **label** |

### 3.3 Apache Spark

The main idea of spark is to speed up batch processing, the iterative algorithm in machine learning, graph processing, and interactive query. Spark can process up to 100 times faster processing than as compared to Hadoop MapReduce [11]. It provides a user-friendly API and shells in Java, Python, Scala, and SQL for writing queries and handling jobs. Intrusion detection has become a critical aspect in the field of cyber security network. Thus, batch processing is an efficient way to process high

volumes of data, which accumulated over a period of time. We have used the hypothyroid CSV file "KDD99 data" in the directory of the distributed Spark as input. The dataset is randomly shuffled and stratify the data (RDD) into four partitions, and an ARFF header with additional metadata attributes is computed using all the CPU cores on our computer.

### 3.4 Parallelizing k-means clustering

Clustering is one of the best-known methods of Unsupervised Learning. Clustering is an algorithm that attempts to find the most natural group using given data. K-means clustering is the most widely known algorithm among these clustering algorithms.
We used PKM clustering to handle abnormal network connections and misplaced data packages which can reduce the overall accuracy of classification. Our PKM algorithm is parallel based on the parallelism of the data inherent above all in the Centroid and distance calculation update operations. The operation of distance calculation can be executed in parallel and asynchronously for each data point.

$$(xi \text{ for } 1 \leq i \leq n) \quad (2)$$

The operation process of PKM Clustering is very simple. First, the algorithm initializes the Centroid of each cluster by selecting K data. And each data is assigned to a cluster of the closest Centroid. Then, the average of new data for each cluster is obtained and designated as a new Centroid. This process is repeated until convergence.
This result can be increased by increasing the number of iterations. The setRuns () function sets the number of clusters to be performed per k, and the setEpsilon () function is a threshold for the difference in the center of the cluster to be checked for each iteration. Adjusting Epsilon affects how much iterations are to be performed per performance of each clustering. Setting Epsilon to a large value will not be sensitive to changes in the cluster center point, and setting it to a small value will also be sensitive to changes in the small cluster center point (sensitivity means doing a new iteration with less variation).

**Algorithm PKM:**

**Input:**

- A collection of instances points
- Four Clusters, K

**Output:**

- K-Centroids
- Every cluster instances

**Steps**

1. define universal Centroid C = <C1, C2, …, CK>
2. Divide instances into subgroups P with the same size
3. For every P1, P2, P3,....Pn. Create a new process
4. Send C to each created process for assigning cluster members and calculating distances
5. Receive cluster members of K clusters from P processes
6. Recalculate new Centroid C''
7. Repeat steps until Centroid doesn't change anymore

**Figure 2.** Parallel K-Means (PKM) Algorithm

```
Cluster 0: 0.74416,0.059244,0,1351.076359,0,0,0,1,1.050972,0,0.050972,0.012345
Cluster 1: 0.74416,0.65,1,1.637371,0,0,0.013099,0,1,0,0,0.278247
Cluster 2: 0.74416,0.65,0,274.606593,0,0,0.051381,1,9.059299,0,0,0.038291
Cluster 3: 0.74416,0.004204,1,0,0,0,0,0,1,1,0,0.981029
Cluster 4: 0.74416,0.019838,0,785.632049,0,0,0,1,6.008581,0,0.03275,0.255657
Cluster 5: 0.024688,0.22712,0,1004.584704,0,0,0,0,494.555069,0,0,0.999182
Cluster 6: 0.74416,0.021342,0,9.656148,0,0,0,0,1.055495,0,0.055495,0.009929
Cluster 7: 0.23115,0.003382,0,43.865614,0,0.024058,0,0,20.222195,0,0.015533,0.201695
Cluster 8: 0.74416,0.019838,0,10347.127328,0,0,0,0.735204,10.130158,0,0,0.469064
Cluster 9: 0.74416,0.65,0,221.096409,0,0,0,1,1,0,0,1

Final cluster centroids:
```

**Figure 3.** The Result of Parallelizing K-Means Clustering Creation

### 3.5 Classification approaches

Machine learning-based Intrusion detection classification provides a good performance and requires less expert knowledge. The main purpose of classification is to build a model from the classified object. The classification accuracy of existing machine learning algorithms needs to be improved because it is difficult to prevent new attacks, as the attackers are continuously changing their attack patterns. For classification evaluation in our model, five different classification algorithms are implemented.

#### 3.5.1 Random Forest

Random Forest is the most representative machine learning algorithm and predictable algorithm of bagging approach. The accuracy of the prediction results (Low Bias) is maintained as the mean value of the decision tree used in the individual prediction models, while the low variance is lowered by the Central Limit Theorem. (The way in which N Decision Trees are voted on). In other words, a random forest is a method of creating multiple decision trees, voting, and determining the result by majority vote.The random forest consists of several decision trees. The random forest has low classification error compared to other traditional classification algorithms[12].

#### 3.5.2 J48 Algorithm

Is a simple statistical classifier based on creating decision tree from training data using the concept of information entropy. The greedy top-down construction technique is applied to induce decision trees for classification. Internal nodes identify the different features, the branches between nodes provide us the possible value these attributes can have in the experiential samples, while the terminal nodes provide us the final value (classification) of the dependent variable[13].

**Algorithm J48:**

**Input:**

- Training Dataset = D

**Output:**

- Decision Tree = T

**Steps:**

- DTBUILD (*D)
- T=φ;
- T= Create root node and label with splitting attribute;
- T= Add arc to root node for each split predicate and label;
- For each arc D= Database created by applying splitting predicate to D;
  - If stopping point reached for this path, then T'= create leaf node and label with appropriate class;
  - Else T'= DTBUILD(D);
  - T= add T' to arc;

**Figure 4.** Decision Tree Algorithm

#### 3.5.3 One'R(One Rule)

It's a straightforward and a very effective classification algorithm commonly used in machine learning applications. It generates a one level decision tree. One'R is able to infer typically simple, yet accurate, classification rules from a set of instances. One'R is also capable of handling missing values and numeric attributes showing adaptability despite the simplicity [14]. The One'R algorithm generates one rule for each record in the training data, and selects rule with a minimum error rate as it's "one rule". To create a rule for the record, the most frequent class for each record value must be determined. The most frequent class is simply the class that appears most often for that recorded value.

**Pseudo-code One'R**

For each attribute A,

a) For each value VA of the attribute, make a rule as follows:

Count how often each class appears

to Find the most frequent class Cf

Create a rule when A=VA

Class attribute value = Cf

b) Calculate the error rate of all rules

Chose the rule with the smallest error rate.

**Figure 5.** One'R Algorithm

### 3.5.4 NaiveBayes Algorithm

Is widely used for a probabilistic classifier based on applying Bayes‟ theorem. The presence or absence of particular features of a class is unrelated to the presence or absence of any other features. Conditional probability is calculated on each feature of the given label and forms a Bayesian network for prediction. Its model is easy to build and fast to predict the class of test dataset. NaiveBayes classifiers perform well in multi-class prediction and many complex real-world situations [15].

$$P(X/C) = \frac{p(x/c)p(c)}{p(x)} \quad (3)$$

$$P(X/C) = P(x1/C) * P(x2/C) *\ldots* P(xn/ C) \quad (4)$$

### 3.5.5 Bagging Algorithm

Is main technique used to combine the algorithms in an ensemble. The algorithm is used sequentially. Bagging algorithm analyzes all the records in the dataset and assigns weights to each of them. The record with a higher value for the weight are the ones that were classified wrongly by the algorithm. Then, the next algorithm receives as an input the dataset as well as the weights for all records in the dataset. The weights allow the algorithm to focus on the records that were the most difficult to classify. These weights are updated according to the results of the second algorithm and the process moves to the third algorithm. This sequence continues until the last algorithm of the ensemble has processed the data. The advantage of this technique is that the most difficult records can be classified correctly without adding too much computational overload. The use of weights, which are updated throughout the process, reduces the computation time as the data goes down the chain of algorithms. In an ensemble using the bagging technique, all algorithm of the ensemble is used in parallel. In this case, each algorithm builds a different model of the data and the outputs of every predictor are combined to obtain the final output of the ensemble [16].

## IV. EXPERIMENTAL ANALYSIS

This section presents the experimental evaluation compare to our integrated model. We trained and tested our models on A real-time KDD’99 data set, 13 features out of 41 are selected based on Information gain algorithm. The experiments were conducted using PKM clustering and the Apache Spark machine learning framework on Intel (R) Core(TM) i5-2430M CPU@ 2.40 GHz and 8 GB RAM running window 7. Further, we have used MATLAB and Weak simulation tool for classification. In this experiment, we have compared the performance of three models based on five intrusion detection algorithms which are designed based on well-known classifiers such as NaiveBayes, Random Forest, J48 Decision trees, Bagging and One'R for detecting abnormal network connections.

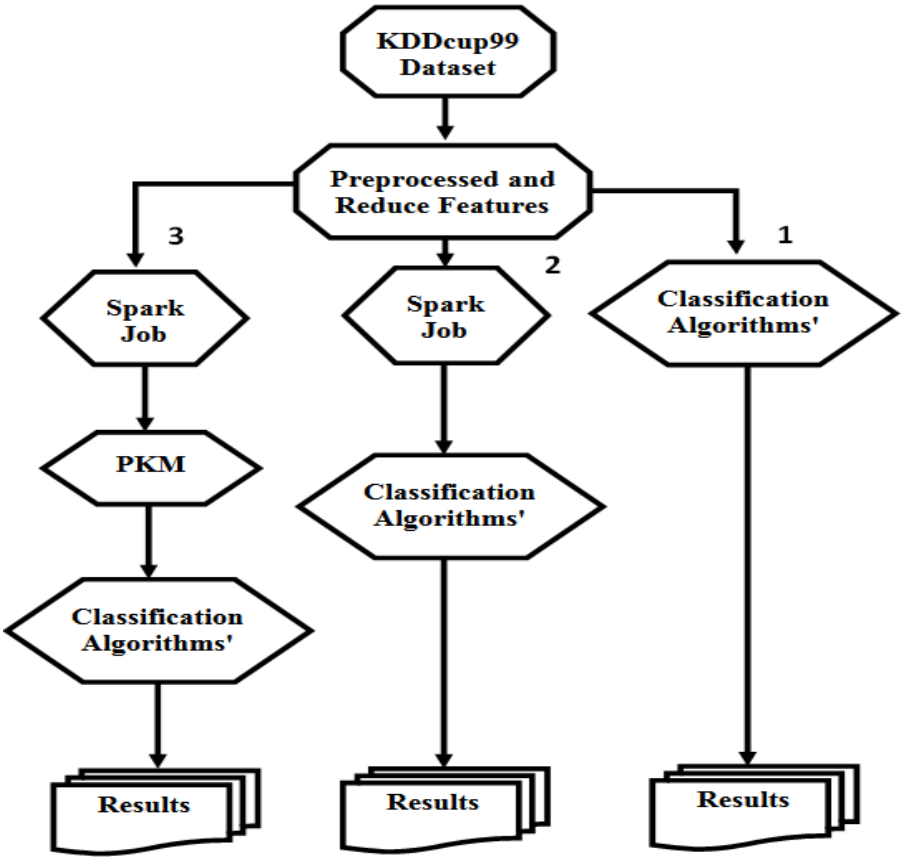


**Figure 6.** Comparison of Three Models

In the first model, we have applied direct classification algorithms based on KDD dataset and selected features. The accuracy of this model was perfect but it takes long prediction time to predict the whole dataset as normal or attack. For this reason, we have applied the second model that consists of spark tool and classification algorithms based on KDD dataset and selected features. The result was reduced by 0.10 % in terms of accuracy and less prediction time compares to the first model. The third model which is integrated model was developing a model of previous two models. That consist of the PKM clustering to handle abnormal network connections and misplaced data packages. The result was perfect in terms of accuracy and prediction time compared to previous two models. The details result from these experiments presented in Table 2 and Figure 7 & 8. In terms of classification Accuracy, Incorrectly Classified Instances and Prediction Time.

**Table 1:** Comparison of Three Different Models and ML Algorithms On KDD'99 Dataset

| **Performance Metrics** | Accuracy **%** | Incorrectly Classified Instances | Prediction Time **M/S** |
|---|---|---|---|
| **Algorithms** / **Models** | **Random Forest** | | |
| 1. **Normal Classification** | 99.97 % | 0.0208 % | 65.44 |
| **2. Spark + Classification** | 99.95 % | 0.0468 % | 03.21 |
| **3. Spark + PK-mean + Classification** | 99.97 % | 0.0256 % | 00.28 |
| | **J48** | | |

| | | | |
|---|---|---|---|
| **1. Models 1 (Normal Classification)** | 99.98 % | 0.0156 % | 17.08 |
| **2. Models 2 (Spark + Classification)** | 99.94 % | 0.0571 % | 01.40 |
| **3. Integrated Models 3 ( Spark + PK-mean + Classification)** | 99.97 % | 0.0238 % | 00.13 |
| | **Bagging** | | |
| **4. Models 1 (Normal Classification)** | 99.96 % | 0.0386 % | 37.04 |
| **5. Models 2 (Spark + Classification)** | 99.88 % | 0.1189 % | 01.10 |
| **6. Integrated Models 3 ( Spark + PK-mean + Classification)** | 99.94 % | 0.0594 % | 00.08 |
| | **OneR** | | |
| **7. Models 1 (Normal Classification)** | 99.41 % | 0.5854 % | 03.77 |
| **8. Models 2 (Spark + Classification)** | 99.33 % | 0.6693 % | 00.13 |
| **9. Integrated Models 3 ( Spark + PK-mean + Classification)** | 99.41 % | 0.5838 % | 00.03 |
| | **NaiveBayes** | | |
| **10. Models 1 (Normal Classification)** | 94.71 % | 5.2818 % | 04.33 |
| **11. Models 2 (Spark + Classification)** | 93.11 % | 6.8818 % | 00.36 |
| **12. Integrated Models 3 ( Spark + PK-mean + Classification)** | 94.49 % | 5.5004 % | 00.13 |

As results are listed in Table 2, we can conclude that the result of our integrated model was perfect in terms of accuracy and prediction time based on five Machine Learning classification algorithms compared with others models.
Figure 7 shows the results three models in terms of Prediction Time.

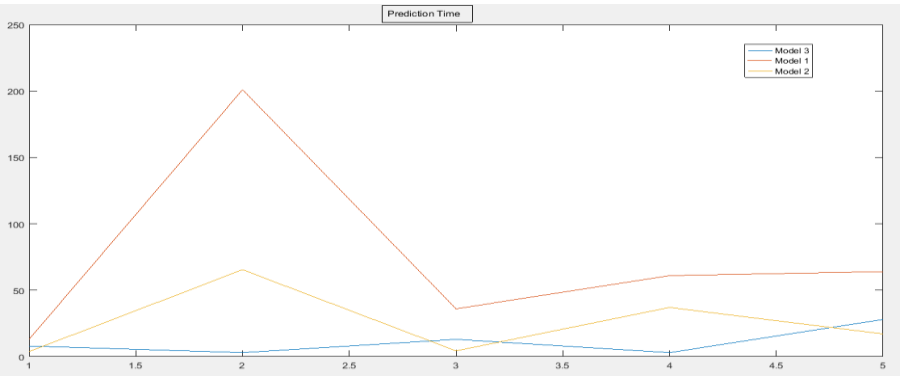


**Figure 7.** Results of Prediction Time for The Three Models and Five Machine Learning Classification Algorithms

Figure 8 shows that the first model and third model result is almost the same in terms of Accuracy and they perform better than the second model. The comparison was made based on five Machine Learning classification algorithms.

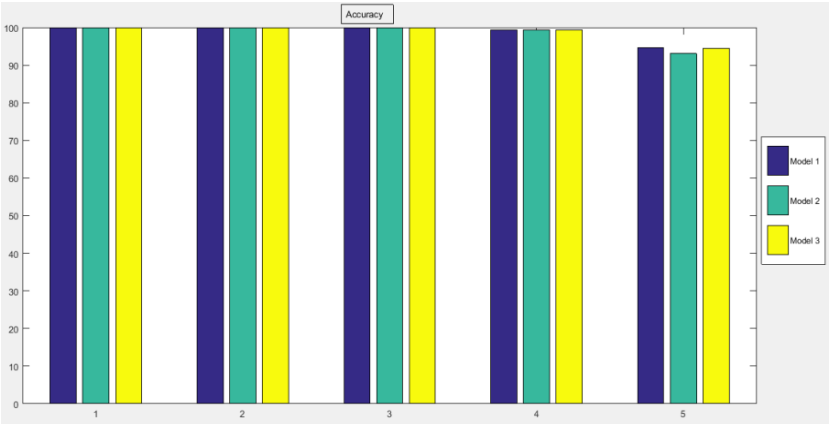


**Figure 8.** Results of Accuracy for the Three Models and Five Machine Learning Classification Algorithms

## V. CONCLUSION

The objective of this experiment was to find model and algorithms which can accurately classify the big data records with less prediction time. The experiment was done using KDD'99 dataset using apache spark and PKM clustering. The comparative results considered in this paper can be used as a baseline for further research. However, more studies should be evaluated, e.g, a different dataset with the heterogeneous number of traffic patterns, and advanced classification techniques for future work.

## REFERENCES

[1] Modi, Chirag N., Dhiren R. Patel, Avi Patel, and Rajarajan Muttukrishnan. "Bayesian Classifier and Snort based network intrusion detection system in cloud computing", 2012 Third International Conference on Computing Communication and Networking Technologies (ICCCNT 12), 2012.

[2] Z. Richard, T.M. Khoshgoftaar, and R.Wald, 2015. "Intrusion detection and Big Heterogeneous Data: Survey." Journal of Big Data 2.1: 1-41, 2015.

[3] Spark, A. Spark Homepage http://spark.apache.org.

[4] Tan Z, Nagar UT, He X, Nanda P, Liu RP, Wang S, Hu J.,"Enhancing Big data security with collaborative intrusion detection", IEEE Cloud Computer, pp. 27–33, 2014.

[5] Gunasekaran, R., et al., Real-Time System Log Monitoring/Analytics Framework.

[6] Solaimani, M., L. Khan, and B. Thuraisingham. Real-time anomaly detection over VMware performance data using storm. in Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on. 2014. IEEE.

[7] Prathibha, P. G., and E. D. Dileesh. "Design of a hybrid intrusion detection system using snort and Hadoop," International Journal of Computer Applications, 73(10), 2013.

[8] S. R. Bandre and J. N. Nandimath, "Design consideration of Network Intrusion detection system using Hadoop and GPGPU," 2015 International Conference on Pervasive Computing (ICPC), Pune, pp. 1-6, 2015. doi: 10.1109/PERVASIVE.2015.7087201

[9] H. D. J. Jeong, W. Hyun, J. Lim and I. You, "Anomaly Teletraffic Intrusion Detection Systems on Hadoop-Based Platforms: A Survey of Some Problems and Solutions," 2012 15th International Conference on Network-Based Information Systems (NBiS), Melbourne, pp. 766-770, 2012. doi: 10.1109/NBiS.2012.139.

[10] Ahmed Fuad Mohammed, Vikas T. Humbe, Santosh S. Chowhan, "Analytical Study of Intruder Detection System in Big Data Environment " Soft Computing: Theories and Applications SoCTA 2016, Volume 2, 2016.

[11] Solaimani, Mohiuddin, Mohammed Iftekhar, Latifur Khan, Bhavani Thuraisingham, and Joey Burton Ingram. "Spark-based anomaly detection over multi-source VMware performance data in real-time", 2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), 2014.

[12] Md. Al Mehedi Hasan, Mohammed NasserB, iprodip Pal, Shamim Ahmad, "Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS)", Journal of Intelligent Learning Systems and Applications, 2014, Volume 6, pp. 45-52.

[13] Ahmed Fuad Mohammed, Vikas T. Humbe, Santosh S. Chowhan, "Data Mining Based Traffic Classification Using Low-Level Features" International Journal of Computer Applications (0975 – 8887),2017.

[14] M. Mazhar Rathore, Anand Paul*, Awais Ahmad, " Hadoop Based Real-time Intrusion Detection for High-speed Networks", 978-1-5090-1328-9/16/$31.00 ©2016 IEEE, vol. 13, no. 7, pp. 1443–1471, July 2016.

[15] Alexandre Balon-Perin, "Ensemble-based methods for intrusion detection", Master thesis for the degree of Master in Computer Engineering Academic year 2011-2012. Norwegian University of Science and Technology (NTNU).

[16] Farhad Alam and Sanjay Pachauri" Comparative Study of J48, Naive Bayes and One-R Classification Technique for Credit Card Fraud Detection using WEKA", Advances in Computational Sciences and Technology, ISSN 0973-6107 Volume 10, Number 6 (2017) pp. 1731-1743.

- **Mr. Ahmed Fuad Mohammed Al-Dubai**

  Research Scholar At School of Computational Sciences,
  S.R.T.M University Nanded, Maharashtra.
  Ahmed_Aldubai86@yahoo.com

- **Dr. Vikas T. Humbe**
  Assistant Professor At School of Technology,
  S.R.T.M University Sub-Campus Latur, Maharashtra.
  vikashumbe@gmail.com

- **Dr. Santosh S. Chowhan**
  Assistant Professor At School of Computational Sciences,
  S.R.T.M University Nanded, Maharashtra.