International Journal of
Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: 10.31695/IJASRE.2019.33579

Volume 5, Issue 10
October - 2019

An Evaluation of Predictive Accuracy of Common Machine Learning Algorithms on Data Stream

OKEKE Chinedu Emmanuel[1], OWOLABI Olumide[2]

Research Scholar[1], Lecturer[2]

[1-2]Department of Computer Science,

University of Abuja, Abuja,

Nigeria

Abstract

Streamed data are a potentially infinite sequence of incoming data at every high speed and may evolve over real-time. This causes several challenges in large scale high-speed data streams in real-time applications. Hence, this field has gained a lot of attention from researchers in recent years. In recent times, there are several areas of human endeavors where data generated are periodical or constantly growing. For instance, capital market, social networks, metrological data, E-commerce, online gaming and betting platforms. This research implemented the common machine learning algorithm on data streams and random data sets to describe the kind of data with more accurate predictions. Data samples were obtained from social media platforms such as Twitter and Instagram within the periods of 4th to 15th February 2019 with a total of 510,738 data samples collected. This is due to the sheer size of these platforms. Random forest, multi-layer perceptron and the k-nearest neighbor algorithms were used to model the data streams using the WEKA and RapidMiner data mining programs. The result of the research shows that Multilayer Perceptron produced the highest level of accuracy in both programs used when compared to the other algorithms used in the research. The findings of this research will be relevant to other researchers willing to develop machine learning tools to test the accuracy of data streams on social media platforms and related fields.

Key Words: *Data Stream, Classifier, Machine Learning Algorithms, Predictive Accuracy, WEKA,, Multilayer Perception, K-Nearest Neighbour (KNN), Random Forest.*

1. INTRODUCTION

There are several areas of human endeavors where data generated are periodical or constantly growing. For instance, capital market, metrological centers, E-commerce, gaming and betting platforms. This kind of data is known as data streams (i.e. data that occur or arrive at interval of time, like seasonal, periodically, yearly, hourly, by the minutes, by the seconds etc.). Data stream basically represents input data that arrives at high rates, stressing communications within the computing infrastructure such that the entire input data cannot be transmitted and stored. Due to frequency of this kind of data, it becomes increasingly difficult to naturally acquire any timely useful information or knowledge needed to make prompt decisions and predictions that could change the course of unfavourable and avoidable trends or improve the overall performance of a system in real time.

For more than two decades now, foremost Industry leaders and researchers have called and strived to build highly reliable information-knowledge based systems that are capable of giving a global status of the data and intelligently make predictions very true for future events. Researchers have proposed several data mining techniques and algorithms such as:- Decision trees, KNN, Mean-Shift, Hoeffdding tree, Naïve Bayes, Neural networks, Genetic algorithm, Fuzzy logic, etc. [1]. These techniques and algorithms do not work effectively for data streams due to frequency and arrival rate of the input data and one time pass constraint. Most importantly, the optimal performances of these algorithms depend on the careful selection of an algorithm suitable for the nature of the data at hand. A dataset may be fully numeric or categorical and choosing an inappropriate algorithm may make the process slow and grossly inefficient due to further level of preprocessing. For instance, adapting K-means, Mean-Shift algorithms for categorical dataset will make the overall performance to degrade when compared to numeric dataset where no further level of processing is required. The common machine learning algorithm amongst other stream mining algorithms developed for data stream mining tend to manage the evolutional changes in stream mining.

The purpose of this research is to experiment with the common machine learning -algorithms on different stream datasets to deduce the class or type of data that produces optimal accuracy of predicting and classifying new sample data.

1.1 Statement of the Problem

There are several areas of human endeavors where data generated are periodical or constantly growing. For instance, capital market, metrological centers, E-commerce, gaming, and betting platforms. Researchers therefore have proposed several data mining techniques and algorithms but most of these techniques and algorithms does not work effectively for data streams due to frequency and arrival rate of the input data and also one time pass constraint. Most importantly, the optimal performances of these algorithms depend on the careful selection of an algorithm suitable for the nature of the data at hand, which may be fully numeric or categorical.

This research work therefore experiments on different stream data sets to describe the data sets that the common machine language algorithm works better with based on accuracy of prediction.

1.2 Aim and Objectives the Study

The aim of this project is to implement the common machine learning algorithms on data streams and experiment on several data sets to describe the kind of data that the common machine learning algorithms work better with for better prediction.

The objectives are as follows:

- To implement the Random Forest, K-Nearest Neighbhour and Multilayer Perceptron Model to train various data streams
- To perform evaluation of the system as to determine the accuracy and efficiency of prediction of new sample data.
- To deduce the class of stream data which the algorithms model above works better on.

2. LITERATURE REVIEW

2.1 Data Mining

Data mining is one of the many vibrant disciplines in artificial intelligence. It is a process of applying computer-oriented methodologies, including new techniques to discover knowledge from data [2]**.** It is simply the analysis of (often large) observational data sets to find unsuspected relationships and to present the knowledge acquired from the data in novel ways that are both understandable and useful to the data owner.

[3] noted that the increased volume of data needed to be extracted from records brought about computer based approach to mining useful information and knowledge from data. Data scientists now leverage on modern technologies of computers networks, and sensors for easier data collection. As data sets have grown in size and complexity, there has been an inevitable shift away from direct hands-on data analysis toward indirect, automatic data analysis using more complex and sophisticated tools.

Data mining now plays a big role in analyzing several billions of records to discover unknown patterns. These discovered patterns help in studying the trends in human areas where time series data constantly collected. Once a subject matter is identified the data scientist will study the particularity of the data and source for a suitable technique or algorithm for the kind of knowledge sought for.

2.2 Data Stream Mining

One of the problems associated with static data mining models as pointed above is the continuous retraining and fitting of models as information feed increases. Whereas data stream is fast arriving and continuous dataset, which calls for special ways for processing and mining knowledge from such transient dataset. Therefore, static models, finite training set and stationary distribution must be completely overhauled. Generally, data stream mining evolved from the concept of data mining.
[4] defined data streams as stochastic processes in which events occur continuously and independently from each another. In other words, the arrival of one data does not hinder the other and its generation is almost non-stop. [5] also define data streams as dataset which continuously and rapidly grow over time.
Data streams occur frequently especially in systems where there is high level of interaction; such as network data dump, access request to a social media platform, data from sensor devices (e.g. temperature sensor), satellite images, metrological data, security report, market dynamics in a capital market etc.

DOI: 10.31695/IJASRE.2019.33579

Due to large volume and frequent arrivals of data streams it is practically unproductive and computationally demanding to take multiple passes of check on data streams or to use a time bounded observation to generalize for many other arrivals. He collectively referred to these constraints as one pass constraint and temporal locality [6].

2.3 Constraints and Research Issues in Data Streams

Preprocessing of data whose analysis will be carried out is of crucial importance, for the usefulness and validity of the derived conclusions in the context of defined research objectives. Data absorbed from existing databases, data warehouses and data marts (i.e. internal and external sources) or from other data sources are usually not in the required or appropriate format for direct input into the data mining algorithms, therefore they consequently inform the need for preprocessing activities, considering the fact that the data quality is a critical factor for successful analysis.

[7] noted that generally, preprocessing activities is based on the application of appropriate methodological procedures for incorporating the temporal dimensions and they include data selection (data sampling); data reorganization (data summarization, complex data manipulation); data exploration (summary statistics, data visualization, OLAP); data cleansing (anomaly detection, noise reduction, missing values analysis, determination of data consistency); data transformation (data recording, data smoothing, data aggregation, data generalization, functional transformation, grouping values, data normalization, feature construction, feature selection).

When preprocessing activities are conducted, strict care should be taken of their association with the area of research, the objectives of analysis, and assumptions underlying data mining methods and techniques. Otherwise, badly preprocessed inputs cause poor quality of output.

[8] made an observation that many of the research studies point out to the fact that the implementation of preprocessing activities takes between 60% and 90% of data miners' total work time on any particular data mining project.

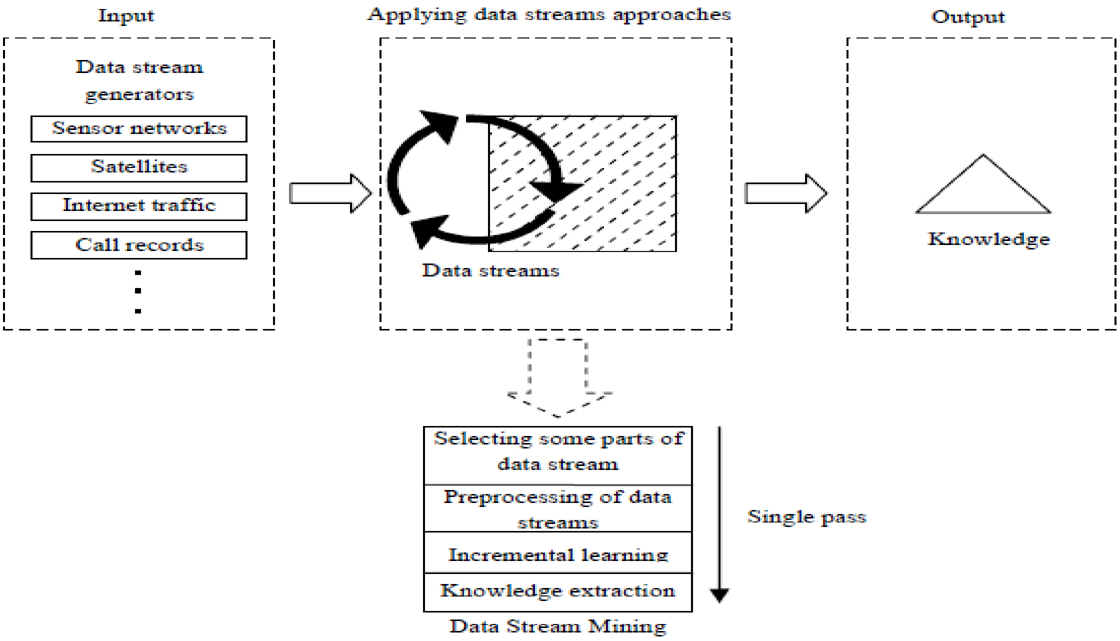


Figure 2.1: General process of data stream mining [9].

2.4 Data Streams Solution Approaches

Data stream solution techniques are divided into two, these are:

- Data Based Techniques
- Task Based Techniques.

2.4.1 Data Based Technique

The data based techniques are data streams solution approaches that make use of a subset of incoming data. It either summarizes the entire dataset or chooses a subset of the incoming stream to be analyzed. Below is a description of data based techniques with pointers to its applications in the context of data stream analysis.

DOI: 10.31695/IJASRE.2019.33579

Table 2.1: Data based Techniques [10]

S/N	TECHNIQUE	DEFINITION	PROS	CONS
1	Sampling	Choosing a data subset for analysis	Error bounds guaranteed	Poor for anomaly detection
2	Load Shedding	Ignoring a chunk of data	Efficient for queries	Very poor for anomaly detection
3	Sketching	Random projection on feature set	Extremely Efficient	May ignore relevant features
4	Synopsis Structure	Quick transformation	Analysis task independent	Not sufficient for very fast streams
5	Aggregation	Compiling summary statistics	Analysis task independent	May ignore relevant features

2.4.2 Task Based Technique

The task based techniques are those methods that modify existing techniques or invent new ones in order to address the computational challenges of stream processing. Approximation algorithms, sliding window and algorithm output granularity represent this category. The task based techniques and their application in the context of data stream analysis are:

Table 2.2: Task based Techniques [10]

S/N	TECHNIQUE	DEFINITION	PROS	CONS
1	Approximation Algorithms	Algorithms with error bounds	Efficient	Resource adaptivity with data rate is not always possible
2	Sliding Window	Analyzing most recent streams	General	Ignores some part of the streams
3	Algorithm granularity	Highly resource aware technique with memory and fluctuating data rates	General	Cost overhead of the resource aware component.

2.5 Machine Learning: Algorithms Types

Machine learning algorithms are organized into taxonomy, based on the desired outcome of the algorithm. Common algorithm types include:

- Supervised learning --- where the algorithm generates a function that maps inputs to desired outputs. One standard formulation of the supervised learning task is the classification problem: the learner is required to learn (to approximate the behaviour of) a function which maps a vector into one of several classes by looking at several input-output examples of the function.
- Unsupervised learning --- which models a set of inputs: labeled examples are not available.
- Semi-supervised learning --- which combines both labeled and unlabeled examples to generate an appropriate function or classifier.
- Reinforcement learning --- where the algorithm learns a policy of how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback that guides the learning algorithm.
- Transduction --- similar to supervised learning, but does not explicitly construct a function: instead, tries to predict new outputs based on training inputs, training outputs, and new inputs.
- Learning to learn --- where the algorithm learns its own inductive bias based on previous experience.

The performance and computational analysis of machine learning algorithms is a branch of statistics known as computational learning theory.

DOI: 10.31695/IJASRE.2019.33579

Machine learning is about designing algorithms that allow a computer to learn. Learning is not necessarily involves consciousness but learning is a matter of finding statistical regularities or other patterns in the data. Thus, many machine learning algorithms will barely resemble how human might approach a learning task. However, learning algorithms can give insight into the relative difficulty of learning in different environments.

2. 6 Data Stream Mining Techniques

Over the years, mining data streams have been a field of interest to the data mining community. The widely used techniques for mining data streams are classified into:

a. Clustering
b. Frequency Counting
c. Classification

2.6.1 Clustering

Clustering involves finding a structure in the collection of unlabeled data; it is simply the process of organizing objects or data into a group based on a particular property or properties. Clustering in data stream mining is one of the most widely studied techniques in the emerging field of data mining.

2.6.2 Frequency Counting

Frequency counting has basically played an important role in various data mining task especially when trying to find interesting patterns from database. The motivation for searching frequent set of data came from the need to analyze transaction data (customer behavior in terms of the purchase product). The first algorithm proposed to tackle this issue is called AIS.

2.6.3 Classification

Classification is a method that represents a set of supervised learning technique where variables that are dependent needs to be predicted based on another set of input variables. The classification techniques follow a process of learning from the class labels of known data classes and then using certain rules it predicts the class of unforeseen data.

2.7 Random Forests

Random forests are built by combining the predictions of several trees, each of which is trained in isolation. Unlike in boosting [11] where the base models are trained and combined using a sophisticated weighting scheme, typically the trees are trained independently and the predictions of the trees are combined through averaging. There are three main choices to be made when constructing a random tree. These are

1. The method for splitting the leafs,
2. The type of predictor to use in each leaf, and
3. The method for injecting randomness into the trees**.**

2.8 Perceptron-based learning

Perceptron is a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector describing a given input. Well-known algorithms based on the notion of perceptron are: Single layered Perceptron, Multilayered Perceptrons and Neural network.

2.9 K-Nearest Neighour Algorithm

The K-Nearest Neighbour Algorithm is the simplest of all machine learning algorithms. It is based on the principle that the samples that are similar, generally lies in close vicinity. K-Nearest Neighbor is instance based learning method. Instance based classifiers are also called lazy learners as they store all of the training samples and do not build a classifier until a new, unlabeled sample needs to be classified. Lazy-learning algorithms require less computation time during the training phase than eager-learning algorithms (such as decision trees, neural networks and bayes networks) but more computation time during the classification process.

Nearest-neighbor classifiers are based on learning by resemblance, i.e. by comparing a given test sample with the available training samples which are similar to it. For a data sample X to be classified, its K-nearest neighbors are searched and then X is assigned to class label to which majority of its neighbors belongs to. The choice of k also affects the performance of k-nearest neighbor algorithm. If the value of k is too small, then K-NN classifier may be vulnerable to over fitting because of noise present in the training dataset. On the other hand, if k is too large, the nearest-neighbor classifier may misclassify the test sample because its list of nearest neighbors may contain some data points that are located far away from its neighborhood. K-NN fundamentally works on the belief that the data is connected in a feature space. Hence, all the points are considered in order, to find out the distance among the data points. Euclidian distance or Hamming distance is used according to the data type of data classes used. In

DOI: 10.31695/IJASRE.2019.33579

this a single value of K is given which is used to find the total number of nearest neighbors that determine the class label for unknown sample. If the value of K=1, then it is called as nearest neighbor classification. The K-NN classifier works as follows:

1. Initialize value of K.
2. Calculate distance between input sample and training samples.
3. Sort the distances.
4. Take top K- nearest neighbors.
5. Apply simple majority.
6. Predict class label with more neighbors for input sample.

Following example shows that there are three classes X, Y and Z as shown in figure 2.2. Now, it is required to find out the class label for data sample P. Here, value of K=5 and the Euclidean distance is calculated for each sample pair and it is found that four nearest neighbor samples are falling in the class label X, while single tuple belongs to class label Z. So, the sample P is assigned to class X as it is the principal class for that sample.

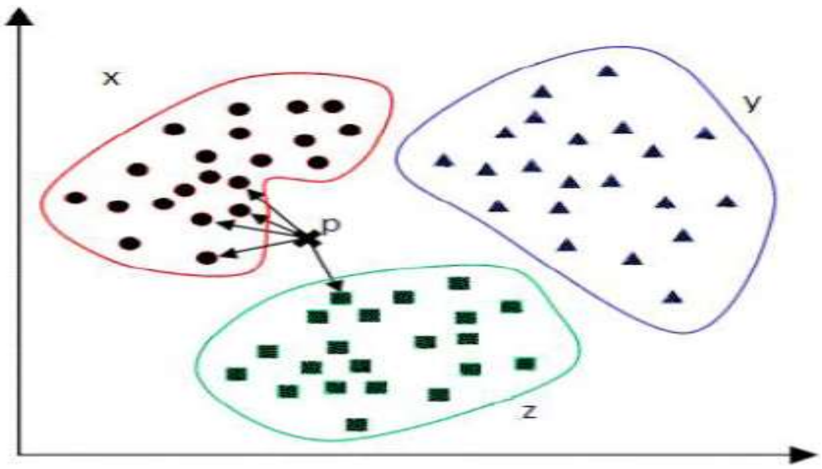


Figure 2.2: An example of KNN Classifier

2.10 Literature Review Summary

From a review of the literature, it is established that a lot of research is still going on different classification algorithms to reduce the error rate and improve the accuracy. Here, the review relates to supervised machine learning methods like Artificial Neural Networks, Decision Trees, Rule Based and Support Vector Machines. The review of literature has driven the focus of the research work in the direction of common machine learning algorithm which gives more regularization, generalization and approximation. As discussed in the performance of the common machine learning algorithm is a critical issue that determines the performance of the classifier.

Table 2.3: Comparison of Performance Metrics of different learning algorithms
(*stars represent the best and * star the worst performance)**

Metrics	Decision Trees	Neural Networks	Naïve Bayes	KNN	SVM	Rule Learners
Accuracy	**	***	*	**	***	**
Learning Speed	***	*	***	***	*	*
Classification Speed	***	***	***	*	***	***
Tolerance to missing values	***	*	***	*	***	**
Tolerance to Irrelevant attributes	***	*	**	**	**	**
Tolerance to Redundant attributes	**	**	*	**	***	**
Highly interdependent attributes	**	***	*	*	***	**
Dealing with different attributes	***	***	***	***	**	***
Tolerance to noise	**	**	***	***	**	*
Tolerance to Overfitting	**	*	***	***	**	**
Incremental learning	**	***	***	***	**	*
Interpretation	***	*	***	**	*	***
Model parameter handling	***	*	***	***	***	***

DOI: 10.31695/IJASRE.2019.33579

3. METHODOLOGY

The methodology employed for this research work is as follows:

i. Methodology
ii. Data collection
iii. Feature Extraction
iv. Data Training or Model Building

3.1.1 Methodology

In this research, four classifiers were selected for comparative analysis. The algorithms are Random forest (RF), K-Nearest Neighbor (KNN), and Multilayer Perceptron (MLP). The decision to choose these algorithms is based on their performance accuracies as reported in the related studies conducted by past researchers. We trained the selected classifiers on the two machine learning (ML) tools using 32 features in the dataset.

3.1.2 Data Collection

The researcher developed a crawler using Twitter and Instagram REST API to collect user data for classifying profiles based on social media DataStream. To the best of my knowledge, Twitter and Instagram privacy policy does not allow researchers to share live streaming data. Thus, abiding by the selected social media REST API rate limit. The data collection covers a period of three weeks from 2nd to 23rd March 2019. A total of 601,435 data with 51,485 unique profiles were collected. Table 3.1 shows the statistics of the data collected. The crawler was integrated with blacklists lookup using PhishTank and Google Safe Browsing APIs. For every stream data that contains URLs, the crawler query PhishTank and Google Safe Browsing to check the status of each URL, whether it is a legitimate or malicious URL. The collected outputs from each API are in JavaScript Object Notation (JSON) format. In total, the dataset contains 35 features, which are discussed in the subsequent section.

Table 3.1: Twitter and Instagram Data set collected using REST API (Source_Survey)

Description of Items	Number of Items
Numbers of stream data	601,435
Numbers of Profiles	51,485
Numbers of URL	198,889
Numbers of users join live streaming	401,823
Number of features	35

3.1.3 Feature Extraction

This is concerned with removal of irrelevant attributes from a large data set. Feature extraction is employed in the analysis of large complex data to reduce the amount of resources required to describe the data set.

3.1.4 Data Training or Model Building

This research adopts the set of features for identifying stream data on social networks and introduces additional features to improve classifier performance. The performance of three machine learning algorithms: Random Forest (RF), K Nearest Neighbor (KNN), and Multilayer Perceptron (MLP) across two popular machine learning tools - WEKA and RapidMiner were evaluated. The above algorithms were used to train the data after which a model is built, this model will be used to test and predict or classify accurately new instances. Often times the major goal of separating data into training set and test set is to help build a classifier with minimum error rate.

3.2 Random Forest Algorithm

Random forest algorithm is a supervised classification algorithm. As the name suggest, this algorithm creates the forest with a number of trees.

In general, the **more trees in the forest** the more robust the forest looks like. In the same way in the random forest classifier, the **higher the number** of trees in the forest gives **the high accuracy** results.

3.2.1 Random Forest Pseudocode:

1. Randomly select **"k"** features from total **"m"** features.
    1. Where **k << m**
2. Among the **"k"** features, calculate the node **"d"** using the best split point.
3. Split the node into **daughter nodes** using the **best split**.
4. Repeat **1 to 3** steps until "l" number of nodes has been reached.
5. Build forest by repeating steps **1 to 4** for "n" number times to create **"n" number of trees**.

DOI: 10.31695/IJASRE.2019.33579

3.3 Algorithm for K-Nearest Neighbor

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry.

3.3.1 The algorithm is as follows:

```
Classify (X, Y, x) // X=training data. Y= Class labels of X, x = unknown sample
for  i = 1 to m do
    Compute distance $d(X_{i,} x)$
end for
Compute set I obtaining indices for the $k$ smallest distances $d(X_{i,} x)$
return majority label for {Y, where i € I }
```

3.4 Multilayer perceptrons

MPs are often applied to supervised learning problems[3]: they train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. Training involves adjusting the parameters, or the weights and biases, of the model in order to minimize error.

3.4.1 Steps of training a multilayer perceptron

Initialization: Assuming no prior information is available, select synaptic weights and threshold value.

Forward Computation: Calculate the induced signal function signals of the network by moving forward through the network layer by layer

Backward Computation: Determine the local gradients of the network

End: Adjusts the weighs still the error rate is significantly reduced

3.5 Experimental Study

The experiment was carried out on two different datasets which the researcher developed a crawler using Twitter and Instagram REST API to observe and ascertain the classifier with optimal accuracy and time of processing. Each of the datasets downloaded has distinct number of instances, attributes and classes. The experimentation procedure is split in 5 steps which are performed on each data set. The procedures used are outlined in the table 3.2.

Table 3.2: Experimental Procedure

Step 1	**Extract features from the datasets**
Step 2	Select data set for training and testing and load it on Weka or RapidMiner system
Step 3	Compute partial and conditional probability of instances of the data sets.
Step 4	Select the instances with maximum conditional probability, and then evaluate the output based on the accuracy
Step 5	Record result and other observations for each data sets
Step 6	Repeat steps 1 to 5 for each classifiers

4. IMPLEMENTATION, RESULTS AND DISCUSSION

4.1 Experiments and Results

An experiment was conducted to evaluate the performance of the selected classifiers using two popular machine learning tools: WEKA and RapidMiner. The evaluation metrics used in this study are accuracy, error rate, Kappa statistic, mean absolute error (MAE), and root mean squared error (RMSE).

4.2 Performance Measures

There are some parameters on the basis of which we can evaluate the performance of the classifiers such as Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbor and Multilayer Perceptron which are explained below.

The **Accuracy** of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier.

The **Error Rate** or misclassification rate of a classifier, M, which is 1-Acc (M), where Acc (M) is the accuracy of M.

The **Confusion Matrix** is a useful tool for analyzing how well the classifier can recognize tuples of different classes.

The **Mean Absolute Error** (MAE) is the average of all absolute errors.

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.

The **sensitivity** and **specificity** measures can be used to calculate accuracy of classifiers. **Sensitivity** is also referred to as the true positive rate (the proportion of positive tuples that are correctly identified), while **Specificity** is the true negative rate (that is, the proportion of negative tuples that are correctly identified). These measures are defined as follows:

$$\text{Sensitivity} = \frac{\text{t-pos}}{\text{Pos}}$$

$$\text{Specificity} = \frac{\text{t-neg}}{\text{neg}}$$

$$\text{Precision} = \frac{\text{t-pos}\ldots\ldots}{\text{t-pos} + \text{f-pos}}$$

where:

t-pos = number of true positives tuples that were correctly classified
pos = number of positive tuples

t-neg = number of true negative tuples that were correctly classified

neg = number of negative tuples

f-pos = number of the false positive tuples that were incorrectly labeled

Thus, it can be shown that the performance accuracy of a classifier is a function of sensitivity and specificity

$$\text{Hence, Accuracy} = \text{Sensitivity}\left(\frac{\text{pos}}{\text{pos} + \text{neg}}\right) + \text{Specificity}\left(\frac{\text{neg}}{\text{pos} + \text{neg}}\right)$$

The above stated performances measures are explain below:

TP Rate: It is the proportion of actual positives which are predicted as positive. The formula is defines as:

$$\text{TP Rate} = \frac{tp}{(tp + fn)}$$ where tp stands for true positive and fn stands for false negative

FP Rate: It is the rate of negatives tuples that are incorrectly labeled. The formula is defined as

$$\text{FP Rate of class Yes} = \frac{fn}{(f_n + t_n)}$$

$$\text{FP Rate of class No} = \frac{fp}{(tp + tp)}$$

Cohen’s kappa statistic is a very good measure that can handle very well both multi-class and imbalanced class problems.

Cohen’s kappa is defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

Where:

p_o is the observed agreement, and p_e is the expected agreement. It basically tells you how much better your classifier is performing over the performance of a classifier that simply guesses at random according to the frequency of each class.

The **Mean Absolute Error** (MAE) is the average of all absolute errors. The formula is:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x_i - x|$$

DOI: 10.31695/IJASRE.2019.33579

Where:

- n = the number of errors,
- Σ = summation symbol (which means "add them all up"),
- $|x_i – x|$ = the absolute errors.

Root Mean Square Error is commonly used in climatology, forecasting, and regression analysis to verify experimental results. The formula is:

$$RMSE = \sqrt{\overline{(f - o)^2}}$$

Where:

- f = forecasts (expected values or unknown results),
- o = observed values (known results).

4. 6 Graphical Representation of Experimental Result on WEKA and RapidMiner

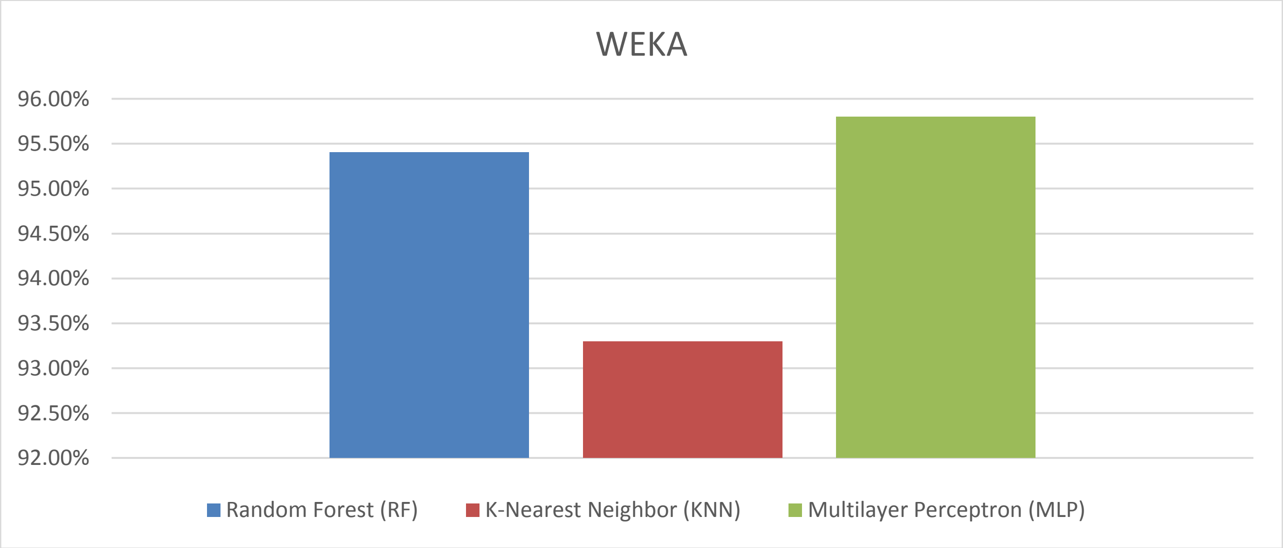


Figure 4.1: Depict the result of the experiment carried out on Data Set with the three algorithms using Weka.

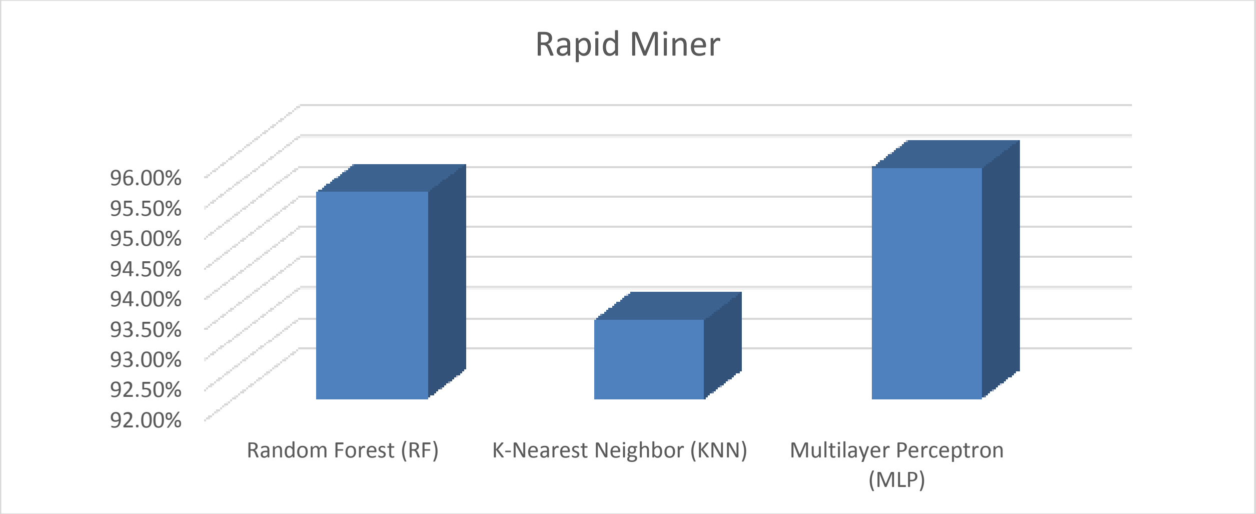
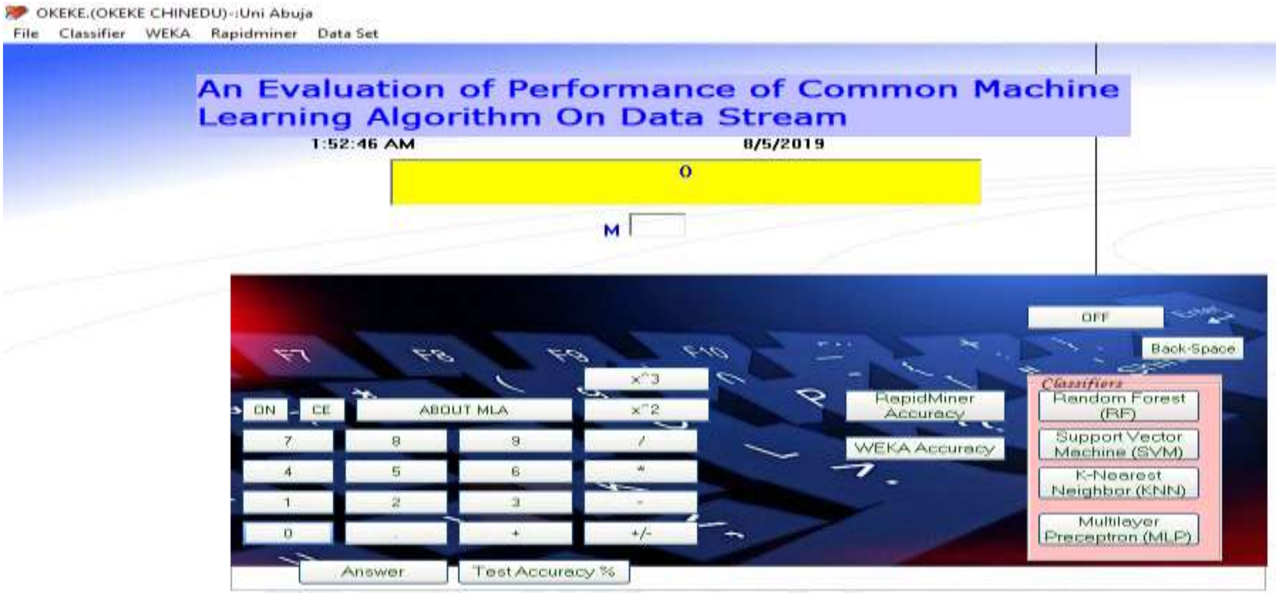


Figure 4.2: Depict the result of the experiment carried out on Data Set with the three algorithms using RapidMiner.

4.6.1 Machine learning Algorithm on Data Stream

The classification accuracy of common machine learning algorithm using Weka and RapidMiner as shown in the Figure 4.3

DOI: 10.31695/IJASRE.2019.33579



OKEKE.(OKEKE CHINEDU)-:Uni Abuja

An Evaluation of Performance of Common Machine Learning Algorithm On Data Stream

1:52:46 AM 8/5/2019

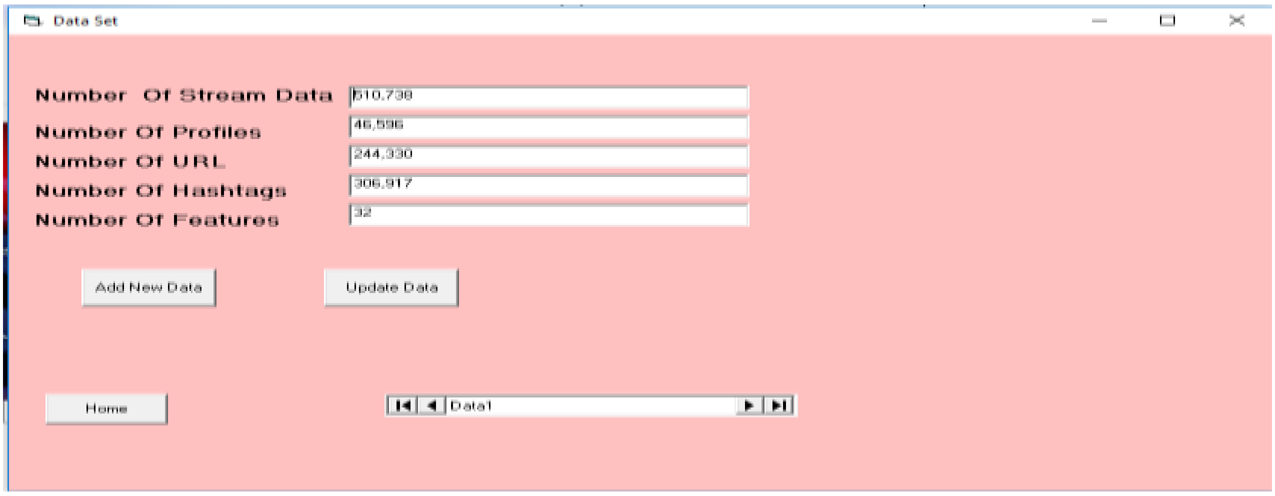
0

M

Figure 4.3: Classification Accuracy Screenshot

4.6.2 Machine learning Algorithm on Data Stream

Dataset used in the experiment is shown in the Figure 4.4



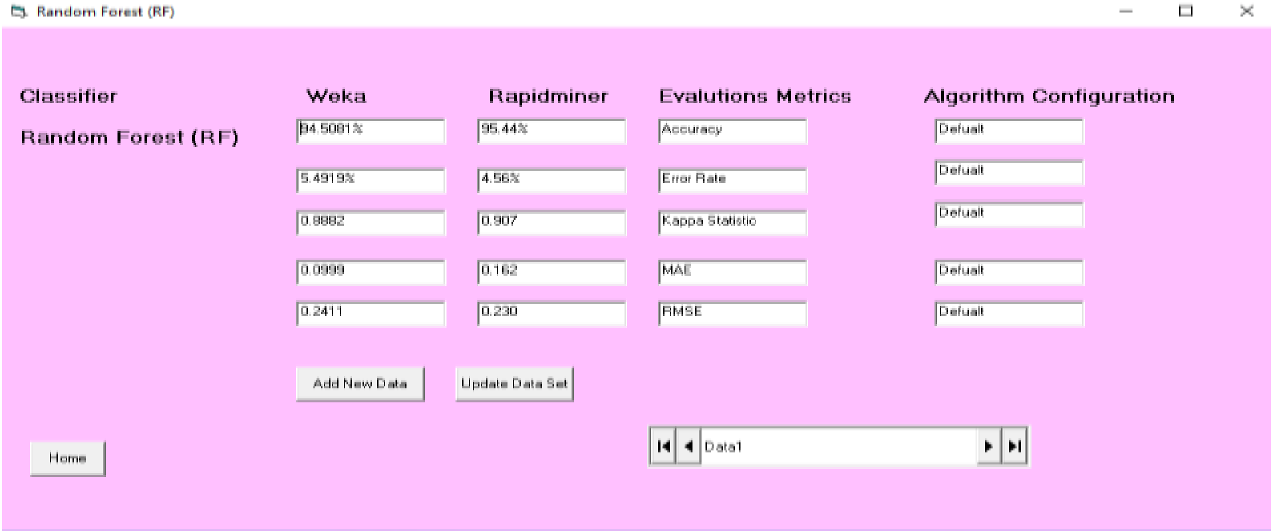
Data Set

Number Of Stream Data	510,738
Number Of Profiles	46,596
Number Of URL	244,330
Number Of Hashtags	306,917
Number Of Features	32

Figure 4.4: Dataset

4.6.3 Random Forest

The classification accuracy of Random Forest Algorithm was tested on the social media data set using Weka and RapidMiner obtained as shown in the Figure 4.5



Random Forest (RF)

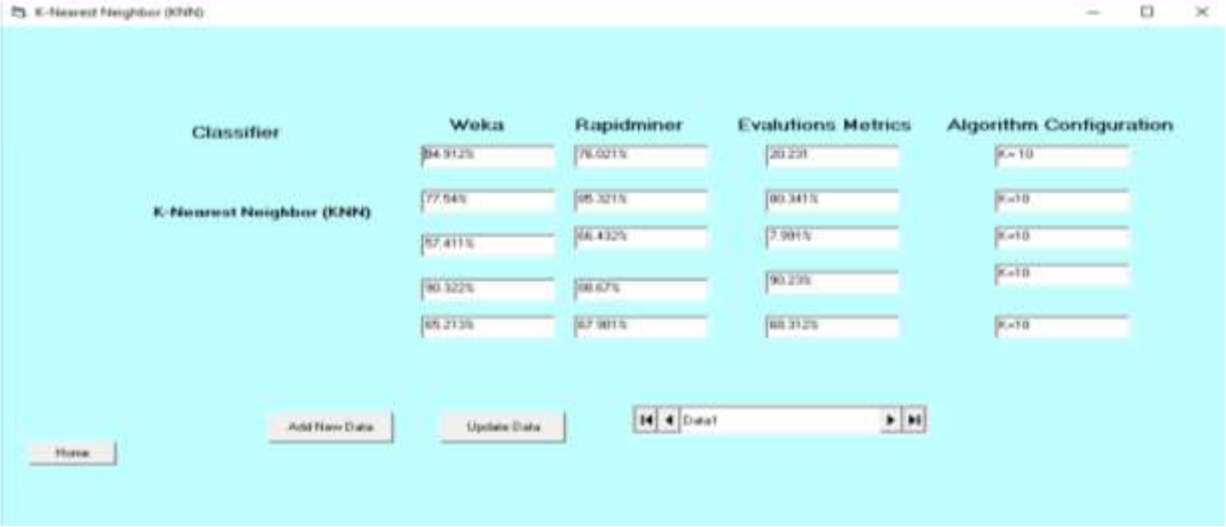
Classifier	Weka	Rapidminer	Evalutions Metrics	Algorithm Configuration
Random Forest (RF)	94.5081%	95.44%	Accuracy	Defualt
	5.4919%	4.56%	Error Rate	Defualt
	0.8882	0.907	Kappa Statistic	Defualt
	0.0999	0.162	MAE	Defualt
	0.2411	0.230	RMSE	Defualt

DOI: 10.31695/IJASRE.2019.33579

Figure 4.5: Classifier of Random Forest (RF)

4.6.4 K-Nearest Neighbour

The classification accuracy of K-Nearest Neighbour Algorithm was tested on the social media data set using WEKA and RapidMiner obtained as shown in the Figure 4.6



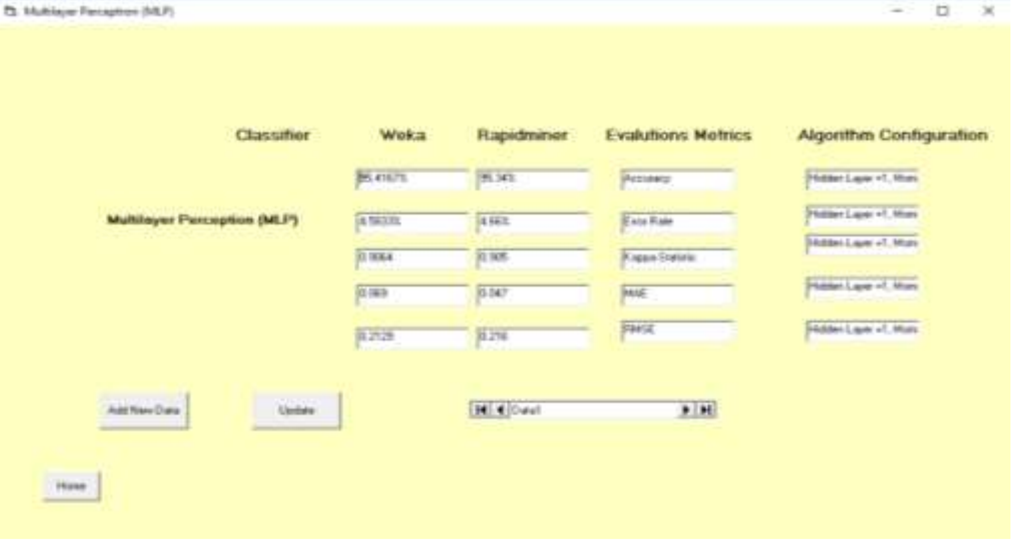
K-Nearest Neighbor (KNN)

Classifier	Weka	Rapidminer	Evalutions Metrics	Algorithm Configuration
K-Nearest Neighbor (KNN)	94.912%	76.021%	20.231	K= 10
	77.54%	95.321%	80.341%	K=10
	57.411%	66.432%	7.981%	K=10
	90.322%	98.67%	90.23%	K=10
	65.213%	67.981%	68.312%	K=10

Figure 4.6: Classifier of K-Nearest Neighbour (KNN)

4.6.5 Multilayer Perceptron

The classification accuracy of Multilayer Perceptron Algorithm was tested on the social media data set using Weka and RapidMiner obtained as shown in the Figure 4.7.



Multilayer Perceptron (MLP)

Classifier	Weka	Rapidminer	Evalutions Metrics	Algorithm Configuration
Multilayer Perception (MLP)	95.4167%	95.34%	Accuracy	Hidden Layer =1, Mom
	4.5833%	4.66%	Error Rate	Hidden Layer =1, Mom
	0.9064	0.905	Kappa Statistic	Hidden Layer =1, Mom
	0.069	0.047	MAE	Hidden Layer =1, Mom
	0.2129	0.216	RMSE	Hidden Layer =1, Mom

Figure 4.7: Classifier of Multilayer Perception (MLP)

5. SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary

This research focused on performance comparison of three selected classifiers (i.e Random Forest, K-Nearest Neighbour, and Multilayer Perceptron) on two popular machine learning tools - WEKA and RapidMiner. We adopted existing features that have been proposed by other researchers to find the accuracy of stream data on social network. In addition, new features were introduced to improve the performance of the selected classifiers. Indeed, the models achieved promising results based on the five performance metrics: accuracy, error rate, Kappa statistic, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Weka classifiers demonstrated good performance in majority of the cases.

6 CONCLUSIONS

The comparison in Table 4.2 above shows that among all classifiers used in the experiment, Multilayer Perceptron classifier produced highest accuracy both in WEKA and RapidMiner when compared with other classifiers. The findings of this research can be useful for other researchers willing to develop machine learning tools to test the accuracy of stream data on social networks.

DOI: 10.31695/IJASRE.2019.33579

6.1 RECOMMENDATIONS

In future, researchers should plan to extend the comparative study by executing Weka algorithms within RapidMiner and to test the cases when reduced feature sets are used to train the selected classifiers. They would also like to compare the performance of their models with existing related studies.

REFERENCES

[1]. Ibe Q. O. (2014). "An evaluation of Predictive Accuracy of Naïve Bayes Algorithm on Data Stream". Computer Science Department, University of Lagos Library.

[2] Berndt Donald. J, James A. McCart, dezon K Finch, Stephen L Luther. (2015) "Data quality in text mining clinical progress notes" ACM Transactions on Management Information Systems (TMIS, Vol. 6,No 1,pp 1.

[3] Konstantinos (2010). "Artificial neural networks as approximators of stochastic processes", Neural Network., Vol. 12, No. 4-5, pp. 647–658.

[4] Gama K. (2010). "Knowledge Discovery from Data Streams". Florida USA: CRC Press.

[5] Charu C. Aggarwal (2010). "On Clustering Massive Text and Categorical Data Streams" Knowledge and Information Systems Article, Vol. 24, No 2: pp 171-196.

[6] Babcock, B., Datar, M. & Motwani, R. (2003). *Load shedding techniques for data stream systems.*. s.l., In Proc. Workshop on Management and Processing of Data Streams.

[7] Kamel. S. Mohamed (2009). "Classification of Imbalanced Data: A Review". International Journal of Pattern Recognition and Artificial Intelligence, Vol. 23, No 4, pp 687-719.

[8] Pyle. D. (1999). "Data Preparation for Data Mining". Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, ISBN: 1558605290-9781558605299.

[9] Kholghi Mahnoosh, &. M. K. (2011) "An analytical framework for data stream mining techniques based on challenges and requirements.". s.l., arXiv preprint arXiv:1105.1950.

[10] Gaber, M., Zaslavsky, A. & Krishnaswamy, S. (2007). "A survey of classification methods in data streams". In: C. C. Aggarwal, ed. *Data Streams: Models and Algorithms,*. New York: Springer, New York, USA.

[11] Schapire, R. and Freund, Y. (2012). "Boosting: Foundations and Algorithms". MIT Press, USA.

DOI: 10.31695/IJASRE.2019.33579