Machine Learning Driven Feature Sensitive Progressive Sampling Model for BigData Analytics

Nandita Bangera, Dr.Kayarvizhy N
BMS College of Engineering
Bengaluru, India

***Abstract*—BigData requires processing a huge data volume, which is an undeniable challenge for academia-industries. The classical sampling techniques are limited when addressing data-imbalance, large data-heterogeneity, multi-dimensionality etc. To alleviate it, in this paper a novel machine learning driven feature sensitive progressive sampling (ML-FSPS) that in conjunction with an improved feature selection and classification environment achieves more than 95.7% of accuracy, even with 10-14% of the original data size. The proposed ML-FSPS model was applied for IoT-device classification problem that possesses exceedingly high data-imbalance, multi-dimensionality and heterogeneity issues. Functionally, the FSPS-driven analytics model at first performed active period segmentation followed by multi-dimensional (descriptive) statistical feature extraction and Wilcoxon Rank Sum Test based feature selection. Subsequently, it executed K-Means clustering over a gigantically huge feature instances ($16,00,000,000$ network traces) Here, K-means algorithm clustered each feature samples into five distinct clusters. With initial sample size of 10%, FSPS model selected same amount of data elements (0.5-5% iteratively) from each cluster for each feature to perform multi-class classification using homogenous ensemble learning (HEL) model. Here HEL encompassed AdaBoost, Random Forest and Extended Tree ensemble algorithms as base classifiers. The simulation results affirmed that the proposed model achieves accuracy of almost 99% even with 10-16% of sample size.**

Keywords*—Feature sensitive progressive sampling; BigData analytics; machine learning; ensemble learning; rank sum test; IoT-device classification

I. INTRODUCTION

The demand for low cost infrastructure in all the business domains has opened up a new horizon for industries to provide decentralized computing solution. Majority of this applications require processing significantly large volume of data to identify patterns and trends to make decisions.[1-3].

To cope up with the demands of the decentralized, data-driven decision support systems, BigData analytics has emerged as one of the most sought technologies [4]. However, BigData which is often characterized in terms of "Volume", Variety", "Velocity", and "Veracity" (say, 4V's) requires computing the gigantically large data to yield decision centric data support [5]. Contrarily the inherent and undeniably unavoidable issues of "Data Heterogeneity", "Unstructured Data" "Multi-dimensionality", and "Unbalanced Data" make most of the existing Big Data analytics methods confined.Majority of the BigData analytics models apply the different machine learning methods [3] to learn over the gigantically large data to perform tasks such as clustering, regression, or classification. However, the efficacy of these methods primarily depends on how effectively they can learn over the large voluminous data in minimum possible time [1-4].

To achieve it, in the last few years different efforts have been made, where the focus is made on improving the process of pre-processing, feature extraction, feature selection, and then classification. However, data being central of these efforts requires a (BigData) computing model to retain "sufficiently small" amount of data to perform analytics without undergoing exhaustive computation and time-exhaustion [2][4][5]. To minimize data load and related computing exhaustion in BigData analytics, authors have found sampling [6] as one of the viable approach. Sampling can not only reduce computing exhaustion but can also retain the minimal data with uncompromising performance [7]Also in the last few years industries have started using or mining a fraction of sample rather than the entire data-warehouse [8]. It improves the scalability as well as timely decision support towards real-time applications [8]. However, the predominant challenge in developing sampling-based approaches originates from the undeniable fact that the occurrence or the frequency of a data element (say, itemset) in a sample might have the different frequency or severity across the complete data set, signifying data imbalance [9]. Under such condition, the classical random sampling approaches might undergo inaccurate performance.

Therefore, alleviating such issue requires identifying optimal size of sample which could provide higher accuracy with minimum possible sample or data load [10]. To cope up with aforesaid demands, recently an approach called progressive sampling has attracted global academia-industries because of its ability to employ minimum data while achieving expected performance [11][12]. The progressive sampling method at first employs minimum data size to perform classification and continues increasing the data volume till it reaches expected level of performance.

However, most of the existing progressive sampling approaches consider random sample selection approach. Random feature selection over exceedingly high data heterogeneity and unbalanced data condition might often results inaccurate performance. It can be because of insufficient or insignificant feature learning. Therefore, a robust computing environment with better pre-processing, feature sensitive feature selection and classification can be a potential analytics solution.

In this paper a futuristic and robust feature-sensitive progressive sampling (FSPS) driven BigData analytics model is proposed. Realizing at-hand analytics problems such as data heterogeneity and unbalanced nature the proposed model inculcates highly efficient pre-processing, feature extraction and selection mechanism, followed by FSPS and heterogeneous ensemble learning model for classification.

II. Related Work

This section highlights some of the important literatures central to progressive sampling in BigData analytics.

Mahafzah et al. [17] proposed a parameterized sampling algorithm for data mining.Authors applied three conditions including the transaction frequency, transaction length and transaction frequency-length to perform sample selection in association rule-based data mining. However, being a multi-phased sampling approach its efficacy over the real-time application remained limited. To alleviate time-complexity in multi-phased sampling, Jia et al. [18] developed an adaptive sampling method that exploited association rule amongst the data elements to select sample size. To improve the performance, authors applied multi-resolution analysis with Shannon sampling theory. Chuang et al. [19] proposed sampling error estimation (SEE) based progressive sampling concept. Here, the key purpose of applying SSE was to estimate the suitable sample size for association rule-based mining. Though, SSE helped achieving sample size without performing association amongst the data elements; however, its efficacy remains suspicious over the realistic large-scale data with higher features and dimensionality. Li et al. [20] applied central limit theorem to estimate the sample size over the large datasets. Unlike other approaches depending on Chernoff bounds, they found their proposed model pragmatic in sync with the association rules mining tasks. Lin [21] examined the associations' lattices on V with a sample V' (a Small chunk or subset). It revealed that merely a very specific kinds of samples possess the "same" association rules with the complete original data and conveys the same meaning. Though, authors intended to exploit homogenous features to retain sample; however, its efficacy due to iterative homogeneity estimation over real-time data traffic becomes suspicious [22]. To resolve this problem, authors in [22] performed association rule mining along with frequent itemset mining to estimate the sample size. Interestingly, this approach selected the sample whose size was independent of both the item-frequency as well as transaction counts. Zhao et al. [23] on the other hand applied hybrid theoretical bound model for frequent itemset estimation, which was later used for sample selection. Moreover, authors applied additive error bound along with the multiplicative error bound to perform sample selection. Exelixis et al. [24] proposed a two-phase sampling-based algorithm, FAST (Finding Associations from Sampled Transactions) for large-scale data mining. In this process, at first an initial sample was selected, which was then processed for support estimation for each selected item in the data to estimate the suitable set of samples. Once selecting the sample authors performed outlier detection by selecting the representative set of items. However, its computational exhaustion can't be denied.

Parthasarathy [25] applied the concept of equivalence with association rule mining to perform progressive sampling [25]. A similar work was done by Thakur et al. [26], who applied association rule approach to estimate the reduced sample size for data mining purpose. Though, unlike [25], authors [26] applied Apriori algorithm to estimate the frequent itemset, and thus exploiting the mid-point itemset it identified the support level across the other data elements. In case the support level of the midpoint itemset is higher in comparison to the user-specific support, that it was selected as a part of sample and its size was increased progressively. Santos et al. [27] applied retrospective sampling over different phases to perform progressive sampling. Similarly, Bosch et al. [28] developed a wrapped progressive sampling concept for large data analysis. Their proposed progressive sampling method employed complete data as input and presented elements in the form of feature vector and labelled each element in one of the known output labels. Thus, it intended to optimise data set by estimating the possible combination of parameter setting by exploiting all possible combinations during training. Realizing data sensitivity in real-time applications Portet et al. [29] developed a multi-phased or multi-period sample selection concept, where authors found that their proposed approach could attain the same performance even with one-third of the original data size. Similar to the work in [26], authors [30] performed itemset partitioning, rather than midpoint itemset estimation. Xeng et al. [31] proposed Bayesian optimization based automatic sample selection method using machine learning. Authors [31] applied machine learning to estimate the hyper-parameters values to estimate the sample size. In fact, it served as a machine learning driven Bayesian optimization for feature selection to estimate sample size. Recently, ElRafey et al. [32] applied machine learning-based progressive sampling approach in which the batch model uncertainty sampling was performed (using semi-supervised machine learning algorithm). Here, the semi-supervised machine learning helped selecting the most significant data points to the sample to perform further learning or classification. However, it failed addressing the key problem of data imbalance and heterogeneity, which is common in BigData analytics.

III. Research Objective

A large number of BigData analytics environment has the major problem of class imbalance which can lead to incorrect predictions and analysis. On the other hand, input data or real-time stream from multiple channels often undergoes heterogeneity with diverse data elements with different or non-uniform significance (towards prediction or decision making). In such cases, merely applying random sampling can't yield optimal performance. This is because a data in sample is not guaranteed to have uniform distribution or frequency across the complete dataset. Similarly, a data element with higher frequency outside the sample is not mandatory to have the same frequency inside the sample. Therefore, in device classification problem, merely applying the random sample would create data imbalance. Also sampling methods employing random sample selection might fail in delivering optimal feature learning and classification (because of data imbalance probability).

Selecting significant feature set from a huge datsaset and progressively sampling the dataset can achieve desired accuracy level and can also reduce the response time factor considering above facts as motivation for the research objective a futuristic, new and robust feature sensitive progressive sampling driven BigData analytics model is developed for IoT-device classification. This proposed model aims to reduce the time required for the analytics and also maintaining the desired level of accuracy.

IV. Proposed System Model

A. System Model

The overall proposed BigData analytics model as shown above in Fig. 1 encompasses the following processes:

1. *Network Traffic Sensitive Active Period Segmentation*

1) Multi-dimensional Descriptive Statistical Feature Extraction

2) Wilcoxon Rank Sum Test based Feature Selection

3) K-Means Driven Feature Sensitive Progressive Sampling

4) HEL-assisted Multi-class Classification.

The detailed discussion of these key functions is given in the subsequent sections.

B. Network Traffic behaviour Assessment and Data Acquisition

In this research, considering the typical cases of data imbalance, multi-dimensionality, data heterogeneity and large-scale data instances, the overall proposed BigData analytics model was designed for IoT-device classification.

Typically, in IoT-ecosystems there can be a large number of independently operating devices connected through a wireless network. Once connected to the wireless-network, the IoT-devices starts generating network traffic called network traces which can of both incoming as well as out coming nature, depending on the type, role, configuration and target-services within the network. IoT-devices within the network perform routine communication with peers and the network gateway or servers. Thus, the communication between the device results network traces or traffic. Though, the different IoT-devices employ varied protocols; however, a majority of such device still use TCP/IP protocols. The overall communication is based on network traffic in which data is generated successively over a time interval comprising the devices, their behavior, operating patterns, etc. Such non-linear network traffic patterns can be analyzed by means of the sophisticated tools such as Wireshark or TCP Dump that at first obtains the traffic packets and analyses the key details (say, traffic behavior or features). Moreover, the tools like packet analyzer operating onto the router can help seeing the incoming and outgoing network traffic, and can generate the traffic records. Here, each record comprises the information within the packet (from the MAC to the application layer of the open system interconnection). Though, in sync with realistic condition, where because of the security protocols such as Secure Sockets Layer (SSL), Transport Layer Security (TLS) and the privacy protection policies of governments, merely the packet header can be employed to perform device classification. However, the key accessible information such as Source ID, Destination ID, Protocol Used, MAC address, Packet size, Transmission Period etc. can be applied to perform more accurate and reliable device classification. These features characterizing device behavior over a definite period have been targeted in this research for device classification [14-16]. In reference to above depicted IoT-network condition, for a large device driven network the corresponding traffic flow can be characterized as per (1).

$$S = \{D^1, D^2, D^3, D^4, D^5, \dots, D^j\} \quad (1)$$

In (1), D^j represents the traffic or the information recorder for the $J-th$ packet. Here, every packet D^j comprises the traffic information and updates as (2).

$$D^j = \{t^j, TLength^j, Protocol^j, eth.Src^j, eth.Dest^j\} \quad (2)$$

In (2), the parameter t^j states the approximate period when the packet is transmitted or received. The other parameter $TLength^j$ states the transmission length, while $eth.Src^j$ and $eth.Dest^j$ represent the source and the destination MAC ID of the devices. The parameter $Others^j$ states the other traffic feature recoded in the $j-$th packet. Noticeably, the above discussed network traffic packets are recoded in the form time-series order, such as, $t^1 < t^2 < \cdots < t^j < \cdots$. Let a network comprising N devices representing $d_1, d_2, d_3, \dots, d_n, \dots (1 \le n \le N)$, the corresponding traffic can be presented as per (3).

$$s = \{D^1_{d_1}, D^1_{d_2}, D^1_{d_3}, D^2_{d_1}, D^2_{d_2}, \dots D^1_{d_n}, \dots\} \quad (3)$$

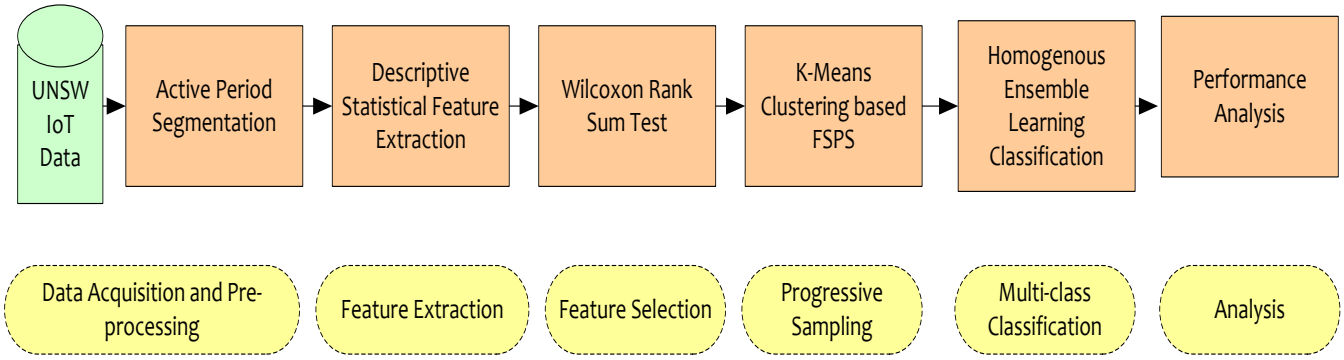

Fig. 1. Proposed ML-FSPS Driven BigData Analytics.

In (3) $D_{d_n}^{i}$ states the $i-$th packet of the IOT-device d_n. Now, towards data pre-processing task, it is needed to extract specific traffic traces or sequences for each IoT-device. In fact, each device type can be classified based on its feature such as MAC address in D or similar, on the basis of the traffic direction. Thus, the device-sensitive packet(s) can be distinguished as per (4).

$$S_{d_n} = \left\{D_{d_n}^{1}, D_{d_n}^{2}, D_{d_n}^{3}, \dots\right\} \quad (4)$$

Observing above discussion, it can be easily inferred that the numerous chunks of information can be logged from the communication traces including Device ID, source and destination ID, timestamp, packet length, protocol etc. Additionally, out of these features other supplementary features too can be derived [34]. Since, a typical IoT network can have a large number of devices having non-linear transmission patterns, identifying active period for each is vital. In other words, in real-time applications, the traffic intensity for the different devices can be different. For instance, a normal CCTV camera can generate almost 140 packets per minutes, on the contrary a motion sensor can generate more than 1900 packets per minute. On the other hand, a CCTV camera can generate the packets 24×7, while a smoke sensor can have significantly lower transmission. It indicates that there are the differences in the active period and their traffic intensity amongst the devices across IoT-ecosystem. The use of average traffic over the observation period can force the model to undergo data-imbalance and hence a machine learning model can show inferior performance. Considering this fact, an active period segmentation is performed.

To achieve active period segmentation, the traffic flow across the defined time-period is segmented into multiple sub-traffics, where only active network traces are considered for the further computation. Though, traffic flow can also be segmented at the interval of time T using time-stamp information. For instance, the network traffic flow from the device d_1 can be segmented after T period, iteratively, as given in (5).

$$S_{d_1} = \left\{Sub_{d_1}^{0-T}, Sub_{d_1}^{T-2T}, Sub_{d_1}^{2T-3T}, \dots, Sub_{d_1}^{tT-(t+1)T}, \dots\right\} \quad (5)$$

In (5), $Sub_{d_1}^{tT-(t+1)T}$ states the all-traffic traces of the device d_1 during iT to $(i+1)T$ time period. Mathematically, it can be presented as per (6).

$$Sub_{d_1}^{tT-(t+1)T} = \left\{D_{d_n}^{t}, D_{d_n}^{t+1}, D_{d_n}^{t+2}, \dots D_{d_n}^{t+m}, \dots\right\} \quad (6)$$

In this paper, MATLAB unique function was applied to segment the network traffic over the time-series information.

C. Multi-dimensional Descriptive Statistical Feature Extraction

In the proposed work, a standard benchmark data named UNSW IoT-traffic traces [13] was considered. Noticeably, the data comprised the network traffic of $24/7$ time-period for 20 days. A total of 1,60,000,000 traces were there as the original data. Unlike, classical methods where the same network traffic is used for prediction or classification, 10-different features for each network trace for each device was obtained. To perform descriptive feature extraction, at first the network traffic over defined time period $Sub_{d_1}^{tT-(t+1)T}$ was split into two broad types; control packets and the use packets. Here, user packet encompassed user-data and device-to-server or gateway communication packets. In this work, a packet was classified for its device to have the protocol either TCP, UDP, HTTP, DNS, ARP, or others. Similarly, on the basis of the direction of packet the traffic can be classified as either transmitted packets or the received packets. Noticeably, features for the different traces characterizing the Device ID, MAC ID, Protocol used, Size of the Data communicated etc were obtained. A specific traffic pattern for example packet size, transmission period or the timestamp etc. can have certain dynamism over the operating periods. Considering this fact a multi-dimensional descriptive (statistical) assessment such as Maximum, Minimum Median, Mean, Variance, Upper-Quartile, Lower-Quartile, Kurtosis, Skewness was performed.

In this manner, a total of ten features including Device Source ID (Packet ID), Source ID, Destination ID, source and destination MAC protocol, IP protocols for both source and destination, packet size, transmission period, etc. Thus, extracting above stated features, a humongous volume of features was obtained. Before proceeding for the sample selection the feature selection algorithm was executed. Here, the WRST algorithm was applied, which is briefed as follows.

D. Wilcox on Rank Sum Test based Feature Selection

The WRST method was used to process the retrieved features in the suggested study. The WRST method is notable for being a sort of non-parametric test with independent samples. This method evaluates the relationship between variables (in this case, network traffic features) and their likelihood of affecting a given device type. WRST was used to estimate the association between network or trace features and their relative inclination towards a given device type in the suggested work. Different extracted attributes were treated as independent variables, whereas device type probability was treated as a dependent variable. This method calculated a p-value for each feature variable based on its importance in device prediction or classification. As a result, each feature factor was classified as significant or unimportant based on its p-value. WRST was applied to each feature element, yielding a collection of characteristics (say, a feature vector) that can be speculated to be the sole important features influencing device type categorization. After obtaining the feature vector the FSPS model was used to select the suitable samples. The proposed FSPS model is described in depth in the next section.

E. K-Means Driven Feature Sensitive Progressive Sampling

The key objective of progressive sampling is to retain the minimum possible nu samples while achieving the expected performance (i.e., accuracy, AUC, etc.). Unfortunately, in majority of the classical progressive sampling methods such as [10], the additional samples are selected randomly, and hence don't consider data imbalance or non-linear nature of the features. Such approaches can greatly be limited due to high inaccuracy. Such random sample selection based progressive sampling methods might select the network traces containing

merely CCTV, or only motion sensor. On the contrary, minimum or possibly negligible frequency of fire sensor traces might skew the learning model towards majority class (i.e., the device(s) with higher packets or its frequency). To alleviate such problems, selecting feature-sensitive samples can be vital. In sync with the proposed IoT-device classification problem, where there is non-linearity in network traces of traffic from each device, random sampling based progressive sampling concept can't be suitable. Considering this fact, in the proposed model, the entire network traces for each feature over the complete operating period (i.e., $24\,hours \times 20\,days$)was clustered . In other words, over a total of 16,000,000 network traces or packets characterizing the different features were clustered using K-Mean algorithm. K-Mean algorithm over the aforesaid packets to cluster entire traces into five distinct clusters (for each feature) was applied. Once clustering the network traces over the aforesaid operating period ($24\,hours \times 20\,days$) the proposed progressive sampling method selected data from each cluster for each feature. The overall process is illustrated in Fig. 2. As depicted in Fig. 2, the proposed FSPS model at first considers 10% of the data (or the selected features) as initial sample, and executes progressive sampling that updates the sample by 0.5-5%, iteratively, till it achieves the expected performance. The sample update takes place as per the model derived in (6).

$$S_i = S_0 + \Delta S_\theta \quad (6)$$

Here, S_i represents the updated sample or data size, while S_0states the initial sample size, which was selected as 10% in this work. The other parameter ΔS_θ represents the progressive addition value, which is selected in between 0.5% to 5%. Here, the value of ΔS_θ is appended iteratively to S_0No, till it results the expected performance (here, accuracy). Noticeably, unlike random selection-based sampling approaches [10], in the proposed FSPS method, samples from each cluster, pertaining to the different features (Fig. 2) was selected. This as a result helped retaining maximum feature diversity to train the model and hence provide better accuracy. Moreover, since the equal samples were taken from each cluster (i.e., K1 to K5 for each feature, as depicted in Fig. 2) to update the data, it tried to avoid data skewness or over-fitting.

F. HEL-assisted Multi-class Classification

A progressive sampling-based analytics model can only be effective if it maintains optimal performance in terms of both sample selection, as well as classification performance. Considering this fact, in this paper, unlike standalone classifiers such as SVM, ANN, decision tree, k-NN etc., a homogenous ensemble learning (HEL) environment was developed. As the name indicates multiple base classifiers of the same category was employed. More specifically, in the proposed HEL model, three different and well-known ensemble classifiers named AdaBoost, Random Forest and Extended Tree classifier, were applied as the base classifiers. Thus, executing these three base-classifiers independently, each device was classified and labelled. The labels obtained by each classifier was applied to estimate the maximum voting ensemble (MVE), and hence with the higher (here, minimum two out of three labels) labels, the MVE model predicted an IoT-device for a specific category. Here, the only motive was to exploit higher consensus for final prediction so as to increase reliability as well as accuracy of the analytics solution.

In the proposed multi-class device classification problem, the following algorithmic paradigm was followed:

1) Let $S = \{1, \dots N\}, C = \{1, \dots, C\}$.

S –Set of traffic instances.

N-Number of traces.

C-Labels for each device.

2) $x = \{x_i, \dots x_n\} \in \mathbb{R}^{N \times D}$ -Input dimensionality.

3) $y = \{y_i, \dots y_n\} \in C$ set of labels for N traces.

4) let $\{X, Y\} = \{(x_i, y_i), \dots, (x_n, y_n)\}$ be the training set, comprising n samples.

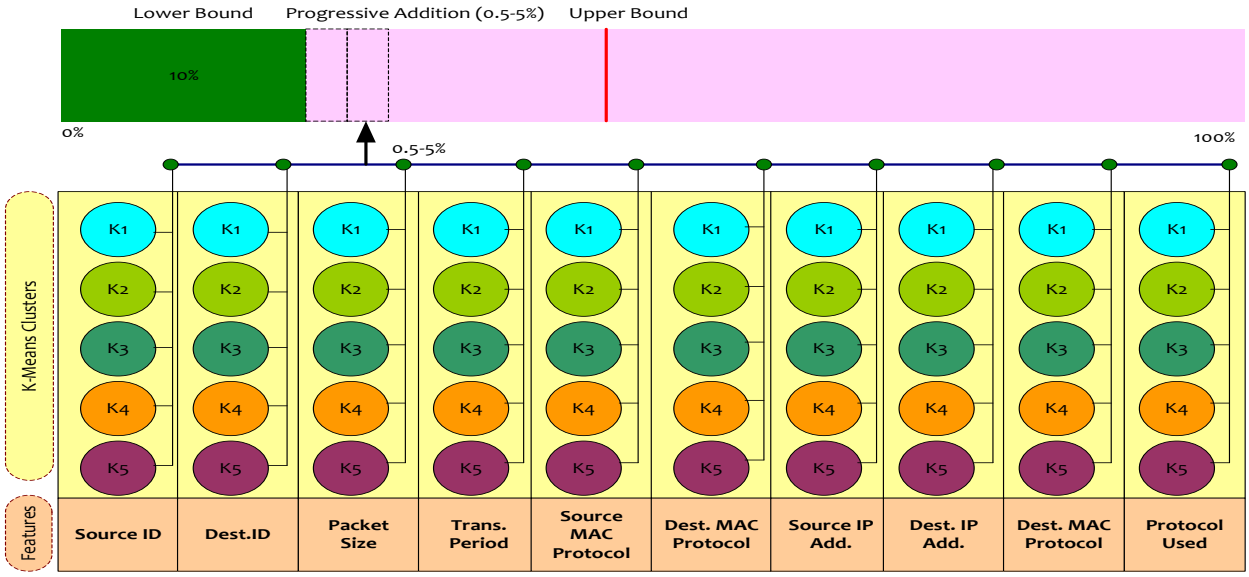

Fig. 2. Proposed Feature Sensitive Progressive Sampling (FSPS) Model.

The motive is to assign label $y_i \in C$ to each network trace $i \in S$ on the basis of the vector x and provide a network trace to the class label y_i .Unlike classical standalone classifier-based learning, three distinct ensemble learning models as the base classifiers was applied. These algorithms are:

1) Random Forest.
2) AdaBoost.
3) Extra Tree Ensemble Classifier.

Noticeably, all these algorithms represent an ensemble learning approach, and thus their use as the base classifier in MVE gives rise to the Homogenous Ensemble Learning (HEL) ability. A snippet of these base classifiers is given as follows.

G. Random Forest Algorithm

The RF algorithm is an ensemble machine learning method that uses numerous tree-structured classifiers. At each input, each tree in the composing tree-structures casts a unit vote (say, an individual vote to establish consensus) for the most likely or popular class. If the number of cases in the training dataset is N, a sample of N cases is randomly chosen from the original data. This sample is also used as a training set for building a tree. If there are M input variables, a number m<M is supplied so that m variables are randomly chosen from the M at each node, and the best split on these m is used to divide the node. During the growth of the forest, the value of m is kept constant .In comparison to the other machine learning models such as SVM, J48, ANN, C5.0, k-NN, etc., RF algorithm requires fewer parameter estimation during processing that makes it more computationally-efficient. In RF algorithm, a collection of distinct tree structured classifier can be defined as (7).

$$\{h(x, \theta_k), k = 1,2, \dots i \dots\} \quad (7)$$

In (7), h states the RF classifier, while $\{\theta_k\}$ refers the random vector distributed identical and each tree possesses a vote for the most probable class at certain input variable x. The nature and dimensionality of θ relies on its use in the tree construction. In RF algorithm the most vital part is the forest of decision trees.. It applies a bootstrapped subset of training samples to train each tree across the constructed forest, which enables almost 70% of the training data usages, while the remaining dataset is stated to be the out-of-bag (OOB) samples, which are typically applied to perform inner cross-validation to assess the classification performance.

In this process during the classification process, the input sample x is classified by going through each tree till a leaf-node is obtained. Here, the classification result (say, the decision function h) is assigned to each leaf node. Thus, the final class label y is estimated by selecting the class with the major votes. Mathematically,

$$y = \text{argm}1,2, \dots, \text{Cax}_{c\in\{\ \}} \sum_{t:h_t(x)=c}^{T} 1 \quad (8)$$

H. AdaBoost

AdaBoost represents an adaptive boosting concept, also referred as a commonplace learning paradigm having the ability to improve the characterization potentiality, iteratively. In initialization the prerequisite tests are doled-out to a similar weight to retrieve some weak learners with some preparation emphases. After each cycle it estimates the error rate of the weak classifier and thus the weight of the accurately classified sample is expanded that reduces the weights of the inaccurately grouped samples. Finally, the weak learner becomes a strong learner to complete the classification. The details of the algorithm applied in this work are given in [33].

I. Extra Tree Classifier

The Extra-Trees classifier constitutes a cluster of unpruned decision trees as per the classical top-down approach. Unlike Random Forest algorithm, it involves randomization of both attribute as well as cut-point selection while splitting a node of a tree. Though, it can also create complete randomized trees possessing structures independent of the output values of the training sample. Primarily, it is distinguishing itself from other tree-based ensemble methods due to two key factors. These are, it splits nodes by selecting cut-points completely at random, and employs the complete training sample (unlike Random Forest which applies bootstrap replica) to enable tree growth. Subsequently, the classified outputs or the predictions of all the trees are combined together to provide final prediction output, by applying MVE method. Summarily, the key concept behind the Extra Tree Classifier is that the complete randomization of the cut-point and attribute altogether with ensemble averaging reduces the variance better in comparison to the weaker randomization approaches used in other methods. Moreover, the use of the original training samples rather than the bootstrap replicas too decreases the likelihood of bias and hence achieved more accurate and efficient classification outputs. Thus, applying above stated classifiers as the base classifiers a MVE ensemble decision was performed where the consensus value was applied to perform device classification. To be noted, since the data considered in this study comprised a total of 26 devices pertaining to six different device categories, the proposed classification model performed multi-class classification. Hence, with the higher number of labels per traffic traces, it labelled the device for the specific category. The simulation results and related inferences are discussed in the subsequent sections.

V. Results and Discussions

Considering the high pace increase in Big Data analytics and its time-efficient computing demands have motivated us to design an optimistically designed computing environment which could achieve expected performance while reducing computational overheads and time-exhaustion. Though to achieve it, the foundation of overall contribution was built onto the improved progressive sampling concept; however, to support efficient computation efforts were made for better pre-processing, feature extraction and selection, and classification as well. Realizing the fact that the use of progressive sampling can help retaining minimum sample volume while achieving higher accuracy, this research employed it as sample selection method. However, recalling the undeniable fact that the typical Big Data analytics models undergo exceedingly high data imbalance, heterogeneity and multi-dimensional features, the random selection based progressive sampling methods can't yield accurate performance. Moreover, the likelihood of

over-fitting and skewed performance can't be ignored. Considering all this facts a feature sensitive progressive sampling (FSPS) model was developed which comprised feature extraction and selection followed by FSPS sampling and homogenous ensemble learning to perform classification.

To assess efficacy of the proposed BigData analytics model, a highly complex and undeniably suitable data pertaining to the IoT-device classification was taken into consideration. A snippet of the considered data is given in the subsequent section. The overall performance analysis was done in terms of classification accuracy, F-Measure and Area Under Curve (AUC). To develop the overall proposed model, MATLAB2020a and Python 3.7 were taken into consideration. Here, MATLAB helped extracting the descriptive statistical features, while rest of the computing algorithms were developed using Anaconda supported Python 3.7 platform. The proposed model was simulated over Microsoft Windows armored with 8 GB RAM and 2.8 GHz processor. The details of the proposed model solution are given in the subsequent section. Before discussing the simulation outputs, a snippet of the data considered and feature distribution is given as follows:

A. Dataset

A benchmark data provided by the University of New South Wales (UNSW), Sydney, Australia [34] was considered. The database was obtained from an IoT-ecosystem created within the university with a total of 26 devices deployed randomly across the university. The network traffic traces were obtained for 20 days (23 Sep. 2016 to 12 Oct. 2016) over $24/7$ operating period. Statistically, the collected data contained a total of 1,60,00,000 network traces or traffic instances carrying packets. The packets captured were parsed to the IP header and was composed to derive other features so as to further perform device category classification or identification. Noticeably, the considered data comprised 26 devices of six different categories. The device and their categories are presented in Table I.

TABLE I. DEVICE CATEGORY

Device Categories with description	No.of Devices	Label/Class
Smart Plugs	5	1
IP Camera	5	2
Motion Sensors	5	3
Temperature Sensor	5	4
Electronics	4	5
Others	22	6

The different devices and their corresponding categories and related labels are given in Table I.

A confusion matrix was obtained in the form of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) to measure the overall performance. Considering data imbalance nature, the classification accuracy, F-Measure and Recall was considered as the key performance parameters. The statistical definition of these performance parameters is given in Table II.

TABLE II. PERFORMANCE PARAMETERS

Parameter	Mathematical Expression	Definition
Accuracy	$\frac{(TN+TP)}{(TN+FN+FP+TP)}$	It is a measure of predicted devices from the overall devices
F-Score	$2.(Recall.Precision)$ $/(Recall+Precision)$	It is harmonic mean of recall and precision numeric values
AUC	-------	It represents the area under curve performance.

The overall performance characterization is made in two phases; intra-model assessment and the inter-model assessment. Here, intra-model assessment discusses the performance of the proposed model with the currently proposed configuration, while the inter-model assessment discusses the relative performance between the proposed FSPS based BigData analytics and other existing algorithms. The outcome of the comparison is elaborated as below.

B. Intra-Model Assessment

In this assessment processes, whether the inclusion of FSPS helps achieving better performance with lower data size was examined. Moreover, the performance with the different sample sizes was also assessed. Additionally, realizing the data unbalanced nature, and a complex multi-class classification problem the accuracy, F-score and AUC with the different base-classifiers was examined. Also, the performance patterns by the proposed model when the sample size is varied were examined. To assess whether the proposed FSPS model helps achieving better performance with minimum data size, the model was tested with 10% data size and subsequently increased sample rate with 0.5%. For the sake of easy presentation and understandability, the results were obtained for 10%, 12%, 14% and 16% of the sample or data size. The accuracy F-score and AUC obtained are given in Table III. Noticeably, here, for classification (over the FSPS samples) the proposed HEL ensemble learning model comprising three base classifiers, Random Tree, AdaBoost and Extra or Extended Tree classifiers were applied.

TABLE III. PERFORMANCE WITH THE DIFFERENT SAMPLE SIZES

Data Size (%)	Accuracy (%)	F-Measure	AUC
10	95.7	0.97	0.99
12	96.4	0.98	0.99
14	97.9	0.98	1.0
16	98.9	0.99	1.0

TABLE IV. PERFORMANCE COMPARISON WITH THE FSPS DRIVEN STANDALONE CLASSIFIERS

Classifier	Accuracy (%)	F-Measure	AUC
Random Forest	97.9	0.98	0.99
AdaBoost	93.6	0.94	0.93
Extended Tree	98.6	0.99	0.99
HEL Ensemble	98.9	0.99	1.0

The key purpose of above assessment (Table IV) was to examine whether the use of FSPS sampling can help a standalone classifier achieving better performance. The results (Table IV) depicts that amongst the different base-classifiers Extended Tree algorithm has exhibited the superior performance with the (multi-class classification) accuracy of 98.6%, F-Measure and AUC of 0.99 and 0.99, respectively. On the other hand, Random Forest algorithm exhibited the accuracy of 97.9%, F-Measure of 0.98 and AUC of 0.99. Unlike Random Forest and Extended Tree algorithm, AdaBoost exhibited inferior with the accuracy of 93.6%, F-Measure of 0.94 and AUC of 0.93. Amongst the three base classifiers AdaBoost algorithm performed inferior; however, recalling the fact that the performance obtained is with merely 16% of the data size, it can be stated as a satisfactory solution.

Noticeably, the proposed IoT-device classification problem was a multi-class classification problem, where the proposed model was supposed to classify each device (here, a total of 26 devices connected to the network and operating autonomously). Though the total number of traces were almost 1,60,000,000, where each trace represents one packet belonging to a specific device of a particular category (Table I). Considering this fact, where the proposed model classified devices into six different categories (it represents the devices of Class 1.0, Class 2.0, Class 3.0, Class 4.0, Class 5.0 and Class 6.0), within micro-average as well as macro-average (between the class and within the class performance, respectively) performance was examined. The ROC performance for each category of the devices after classification was tested. The results obtained by the proposed FSPS-driven HEL ensemble classifier is given in Fig. 3.

Observing the result (Fig. 3), it can observed that the proposed model has obtained the AUC near 0.98 for the complete classes, while the AUV observed for each class (macro-average ROC) is also 0.98. For multi-class classification as well, the average AUC obtained is 0.98.

Typically, in progressive sampling based BigData analytics, in addition to the accuracy performance, time-efficiency too remains the key motive to meet VELOCITY demands. In this reference, relative time-efficiency in between the original data (ORIG) and the FSPS based selected data (PSAM) was compared. The results obtained are given in Fig. 4 and Fig. 5 As depicted in Fig. 4, the proposed progressive sampling-based model (PSAM) performs significantly lower computation time (in seconds) in comparison to the original data-based analytics. Undeniably, such efficacy could be contributed due to significantly reduced data size (almost 86%). It indicates the robustness of the proposed model towards real-time BigData analytics, even under multi-class classification demands.

C. Inter-Model Assessment

In this section, the performance by the proposed model is compared with the other approaches. However, the survey indicates a few such as the work by ElRafey et al. [32] who developed a hybrid active learning based progressive sampling method. More specifically, authors developed a Progressive Batch Model Uncertainty Sampling (PBMUS) model to increase sample size proactively to cope up with (performance) demands. Authors simulated their model with the different datasets, including synthetic data as well as the real-time data. They applied Decision Tree C5.0 algorithm for classification. Authors examined their performance in terms of the classification accuracy and AUC considering 50% of the data size, while the increment boundary was decided as 1%.

Venkatpathy et al. [30] too examined the efficacy of progressive sampling methods with real-time data. Though, the data considered in [30] were smaller in size and diversity as is expected from the Big Data analytics, to assess relative performance, we have considered it as a reference work, as well. Authors [30] had applied Apriori information to estimate the most frequent itemsets and resulting mid-point itemset for association rule-based mining. Authors have examined their performance with the datasets like Mushroom, Chess, Connect, Retail data, Traffic accident data, and synthetic data. To perform relative comparison, the average performance by [30].was calculated. Summarily, the performance comparison of both models and the proposed model is tabulated in Table V.

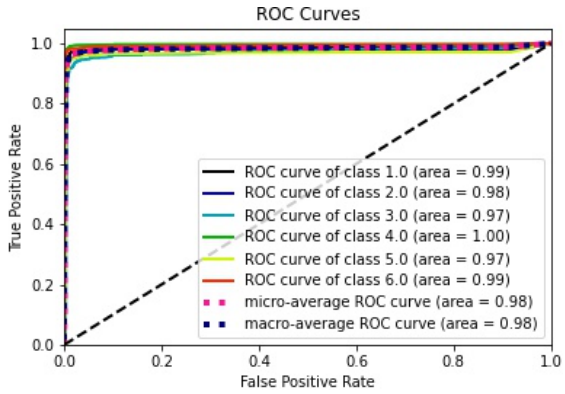

Fig. 3. ROC Performance by the Proposed FSPS-driven HEL Ensemble Model.

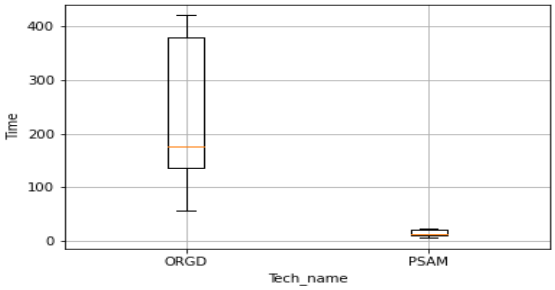

Fig. 4. Time Performance Analysis.

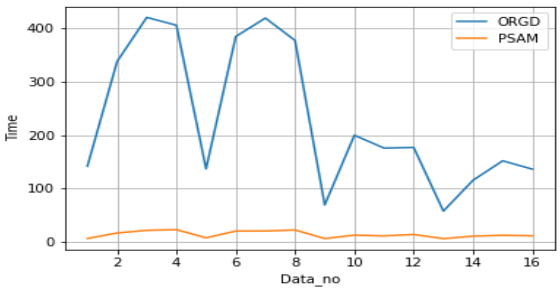

Fig. 5. Time Comparison of Original Data and Progressive Sampled Data.

TABLE V. INTER-MODEL COMPARISON PERFORMANCE OF PROGRESSIVE SAMPLING METHOD

Technique	Accuracy (%)	AUC
[30]	78.0	-
[32]	79.9	79.2
Proposed	98.9	1.0

The above results affirm that the proposed FSPS model achieves significantly better performance than the other state-of-art (progressive sampling) approaches.

Recalling the problem of IoT-device classification, the performance of the proposed model with other state-of-art methods such as [34] was examined. Bai et al. [34] applied the same dataset of UNSW to perform device classification. The authors merely applied the LSTM-CNN as classifier to perform classification over average features. The highest classification accuracy obtained by authors [34] could be merely 74.8%, which is significantly lower than the proposed model. To be noted, authors [34] had applied the complete data size (almost 2.7 GB) to perform classification. On the contrary, in the proposed model FSPS enabled applying merely 10%-16% of the original data to perform classification. Authors in [34] stated that their proposed LSTM-CNN based model could achieve the accuracy of near 99% with 75% of the data size, while with 25% of the training data they could achieve the maximum of 88.2%. However, these performances were merely for the two-class classification. For the multi-class classification, which is expected from the IoT-device classification problem (Table I), the average performance over five repeated simulation was 74.8%, which is significantly lower than the proposed model. Noticeably, in [34] authors also examined the different machine learning classifiers for their respective efficacy for device classification, and hence have compared the performance of the proposed model with the existing approaches [34].

TABLE VI. INTER-MODEL PERFORMANCE COMPARISON FOR IOT- DEVICE CLASSIFICATION

Reference	Technique	Accuracy (%)
Existing work	Support Vector Machine	58.5
	Random Forest	30.1
	KNN	27.6
	Decision Tree	46.4
	AdaBoost	48.5
	LDA	49.4
	QDA	52.4
	Multilayer perceptron	52.1
	Convolutional Neural Network (CNN)	56.3
	Long- and Short-Term Memory (LSTM)	65.4
	LSTM-CNN	74.8
Proposed work	Random Forest	97.9
	AdaBoost	93.6
	Extended Tree	98.6
	HEL Ensemble	98.9

The results depicted in Table VI shows that in comparison to the existing IoT-device classification systems, the proposed (FSPS-driven HEL ensemble learning) model exhibits superior even at significantly lower sample or data size.

VI. CONCLUSION

This paper primarily focused on developing a feature sensitive progressive sampling (FSPS) approach which could retain optimal performance even with minimal data size. Moreover, the key emphasis was to inculcate FSPS while addressing the key problem of data imbalance, multi-dimensionality and data heterogeneity in BigData analytics. Recalling the fact that in BigData analytics merely sampling can't guarantee the optimality of the performance and hence improving both data as well as computing environment is must, this research improved each functional component of the analytics solution. Unlike random (sample) selection based progressive sampling methods, which can't address the problem of data-imbalance, the proposed model employed machine learning driven FSPS to retain maximum possible feature diversity to perform better learning and hence classification performance.. The simulation results exhibited accuracy of 98.9%, F-score of 0.99 and AUC of more than one, affirming robustness of the proposed model towards lightweight, time-efficient and reliable BigData analytics solution. In future the focus can be made on further reducing data imbalance likelihood by applying certain re-sampling concepts. In addition, in future other machine learning models can also be assessed to have better performance for a generalized solution.

REFERENCES

[1] O. Duda et al., "Data Processing in IoT for Smart City Systems," 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Metz, France, 2019, pp. 96-99.

[2] Thibaud Chardonnens, "Big Data analytics on high velocity streams: specific use cases with Storm", Software Engineering Group, Department of Informatics, University of Fribourg, Switzerland, 2013.

[3] H. Chiroma et al., "Progress on Artificial Neural Networks for Big Data Analytics: A Survey," in IEEE Access, vol. 7, pp. 70535-70551, 2019.

[4] R. A. Alshawish, S. A. M. Alfagih and M. S. Musbah, "Big data applications in smart cities," 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, 2016, pp. 1-7.

[5] P. Bellini, F. Bugli, P. Nesi, G. Pantaleo, M. Paolucci and I. Zaza, "Data Flow Management and Visual Analytic for Big Data Smart City/IOT," 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and SmartCityInnovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/ IOP/SCI),Leicester, United Kingdom, 2019, pp. 1529-1536.

[6] Cochran W.G., Sampling Techniques, 3rd edition, John Wiley and Sons, New York, 1977.

[7] Parthasarathy S., Efficient Progressive Sampling for Association Rules, In: Ohsuga S. (Ed.), Proceedings of the IEEE International Conference on Data Mining (9-12 December 2002, Maebashi City, Japan), IEEE Computer Society, 2002, 354-361.

[8] Chen B., Haas P., Scheuermann P., New Two-Phase Sampling Based Algorithm for Discovering Association Rules, In: Zaki M.J. (Ed.), Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (23-26 July, 2002, Alberta, Canada), ACM, 2002, 462-468.

[9] Zaki M.J., Parthasarathy S., Li W., Ogihara, M., Evaluation of Sampling for Data Mining of Association Rules, Proceedings of the 7th

International workshop on Research Issues in Data Engineering (7-8 April 1997, Birmingham, UK), IEEE Computer Society, 1997, 42-50.

[10] N. Bangera, and N. Kayarvizhy, "A Progressive Sampling based Approach to Reduce Sampling Time", 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT-2019), MAY, pp. 74-78.

[11] Chuang K.T., Chen M.S., Yang W.C., Progressive Sampling for Association Rules Based on Sampling Error Estimation, LECT NOTES COMPUT SC, 2005, 3518, 505-515.

[12] Estrada A., Morales E.F., NSC: A New Progressive Sampling Algorithm, Proceedings of the Workshop: Machine Learning for Scientific Data Analysis (Iberamia) (22-26 November, 2004, Iberamia), 2004, 335-344

[13] A. Hsu, J. Tronty, D. Raymondy, G. Wang, A. Butt, "Automatic IoT Device Classification using Traffic Behavioral Characteristics", IEEE Conference, 2019, pp. 1—7.

[14] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "Iot sentinel: Automated device-type identification for security enforcement in iot," in Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on. IEEE, 2017, pp. 2177–2184.

[15] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Characterizing and classifying iot traffic in smart cities and campuses," in Computer Communications Workshops (INFOCOM WKSHPS), 2017 IEEE Conf.on. IEEE, 2017, pp. 559–564.

[16] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized IoT devices using machine learning techniques," CoRR, vol. abs/1709.04647, 2017. [Online]. Available: http://arxiv.org/abs/1709.04647.

[17] Mahafzah B.A., Al-Badarneh A.F., Zakaria M.Z., A new sampling technique for association rule mining, J INF SCI, 2009, 35, 358-376.

[18] Jia C.Y., Gao X.P., Multi-scaling sampling: an adaptive sampling method for discovering approximate association rules, J COMPUT SCI TECHNOL, 2005, 20, 309-318.

[19] Chuang K.T., Chen M.S., YangW.C., Progressive Sampling for Association Rules Based on Sampling Error Estimation, LECT NOTES COMPUT SC, 2005, 3518, 505-515.

[20] Li Y., Gopalan R.P., Effective Sampling for Mining Association Rules, LECT NOTES COMPUT SC, 2005, 3339, 391-401.

[21] Lin T.Y., Sampling in Association Rule Mining, In: Dasarathy B. (Ed.), Data Mining and Knowledge Discovery: Theory, Tools, and Technology VI, Proceedings of SPIE (Orlando, Fl, USA), SPIE, 2004, 161-167.

[22] Chakaravarthy V.T., Pandit V., Sabharwal Y., Analysis of sampling techniques for association rule mining, In: Fagin R. (Ed.), Proceedings of the 12th International Conference on Database Theory (23-25 March 2009, St. Petersburg, Russia), ACM Press, 2009, 276-283.

[23] Zhao Y., Zhang C., Zhang S., Efficient frequent itemsets mining by sampling, In: Li Y. (Ed.), Proceedings of the fourth International Conference on Active Media Technology (7-9 June, 2006, Amsterdam, The Netherlands), IOS Press, 2006, 112-117.

[24] Chen B., Haas P., Scheuermann P., New Two-Phase Sampling Based Algorithm for Discovering Association Rules, In: Zaki M.J. (Ed.), Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (23-26 July, 2002, Alberta, Canada), ACM, 2002, 462-468.

[25] S.Parthasarathy, "Efficient progressive sampling for association rules", IEEE International Conference on Data Mining, 2002.

[26] S. S. Thakur, Shalini Zanzote Ninori, "An Improved Progressive Sampling based Approach for Association Rule Mining International Journal of Computer Applications" (0975 –8887), Volume 165 – No.7, May 2017.

[27] P.A. De los Santos, R.J. Burke, J.M. Tien, "Progressive random sampling: A multiperiod estimation technique with applications IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews", Volume: 30, Issue: 4, Nov 2000.

[28] Antal van den Bosch, "Wrapped Progressive Sampling for optimizing Learning Algorithm Parameters, Netherlands Organisation for Scientific Research".

[29] François Portet, Feng Gao, Jim Hunter and René Quiniou, "Reduction of Large Training Set by Guided Progressive Sampling: Application to Neonatal Intensive Care Data".

[30] Venkatapathy Umarani Muthusamy Punithavalli- Analysis of the progressive sampling-based approach using real life datasets https://link.springer.com/journal/13537.

[31] Zeng X, Luo G, "Progressive sampling Based Bayesian optimization for Efficient and Automatic Machine Learning Model Selection", Springer 2017.

[32] Amr ElRafey and Janusz Wojtusiak, "A Hybrid Active Learning and Progressive Sampling Algorithm, International Journal of Machine Learning and Computing", Vol. 8, No. 5, October 2018.

[33] Q. Li, W. Li, J. Wang and M. Cheng, "A SQL Injection Detection Method Based on Adaptive Deep Forest," IEEE Access, vol. 7, pp. 145385-94, 2019.

[34] L. Bai, L. Yao, S. alil, S. Kanhere, X. Wang, and Z. Yang, "Automatic Device Classification from Network Traffic Streams of Internet of Things", 2018, IEEE 43rd conference on Local Computer Nrtworks (LCN), 2018, pp. 1-9.