International Journal of Innovation, Creativity and Change. www.ijicc.net
Volume 16, Issue 2, 2022

ICSM: Imbalanced Chunk-Based Stream Model

Mashaal A. Alfhaid[1], Manal A. Abdullah[2], [1]Faculty of Computing and Information Technology, King Abdul-Aziz University, Jeddah, Saudi Arabia, ORCID: 0000-0002-5349-0248, [2]faculty of Computing and Information Technology FCIT, King Abdulaziz University KAU, Saudi Arabia SA. ORCID: 0000-0003-2660-6011

Stream data mining becomes one of the major important topics. It is considered one of the fields that face challenges due to continuously arriving data that processed at a single scan. As pre-processing is one of the critical stages in data mining, imbalanced stream data gain significant popularity in the last few years among researchers as many real-world applications suffer from this issue. Handling imbalanced data is mandatory for more accurate and reliable learning models and most importantly have a fast-running time. In this paper, an Imbalanced Chunk-based Stream Model (ICSM) is proposed. It monitors the overall imbalanced ratio of the classification over binary classes. ICSM tested based on several factors: Imbalanced-ratio (IR) diversity by using ten generated data streams, over-sampling techniques and base-classifiers quality. Moreover, it is compared against state-of-art algorithms such as Over/under ensemble OUSE. The experiments show that the ICSM cannot be proceed with too small chunk-size. Nevertheless, large chunk-size can affect the time-delay. It has also been proven that SVMSMOTE outperform other sampling techniques in highly imbalanced ratio. While the GUSSIAN classifier have better performance comparing to other classifiers. Lastly, ICSM outperforms other state-of-art techniques in term of evaluation metrics and time delay in about 95%.

Keywords-: *Data Mining, Imbalanced data, Data Stream Classification, Data preprocessing, Ensemble-based.*

1. Introduction

Nowadays Internet of things (IoT) became one of the trending topics that have many real-world applications such as weather forecasting, stock market, social media, healthcare and many more (Junyi Li, Xintong Wang, Yaoyang Lin, Arunesh Sinha, & Michael Wellman, 2020; Sherali Zeadally & Oladayo Bello, 2021). These applications generate huge data that should be analyzed online. In the streaming data mining field and knowledge extraction, dealing with this type of data requires other techniques and methods than traditional static data, which need more speed and less complexity. As data pre-process is one of the important phases in data mining and knowledge extraction that consumes most of the time in the project (Chang-Woo Song, Hoill Jung, & Kyungyong Chung, 2019). Accurate data processing leads to the extraction of the correct information and a better accuracy model. One of the pre-processing steps is dealing with imbalanced data, which have become one of the major trending topics nowadays especially in streaming data type. As dealing with a disproportion between classes is important in the decision-making process. Moreover, learning from non-stationary data streams remains the focus of intense research, since many real decision-making problems should process on streaming data where not many studies had been considering the imbalanced streaming data as static imbalanced data. Therefore, when dealing with imbalanced streaming data several challenges needed to be considered which have been mentioned in our previous paper (Mashaal A Alfhaid & Manal Abdullah, 2021) such as single scan, concept drift, timing issues. In this paper, an adaptive learning model that deals with imbalanced streaming data called the Imbalanced chunk-based Stream Model (ICSM) had been proposed and evaluated. The main aim of this paper is to enhance the performance in handling the problem of binary imbalance classification and in-memory problem in streaming data. Where this aim will achieve by provide a flexible and effective method that will increase the recognition rate of minority class occurrences with the least time and complexity required. The reason for focusing on the minority or so-called rare events is because many applications in real life require a model that correctly predicts it and doesn't bias toward the majority which is not the focus of these kinds of applications. One of these applications is fraud detection where the main aim is to predict the rare fraud detection not the opposite. The ICSM model will process the streams of data as chunks that include binary classes and track the imbalanced ratio degree in each chunk to solve the imbalanced problem in order to produce a more optimal model.

The paper is organized as follows. Section 2 contains an overview of related work regarding the imbalanced data stream classification methods. Section 3 describes the proposed Imbalanced Chunk-Based Stream Model (ICSM). Section 4 Includes the experiments plan with the description of data sets. Results as well as the analysis of the results, and lessons learned can be found in section 5. The last section 6 concludes the work.

2. Related Work

A lot of research has been published that related to the imbalanced data stream in terms of static data as mentioned in our previous paper (Mashaal A Alfhaid & Manal Abdullah, 2021). However, few considered the imbalanced topic related to the stream data field. As most of the work focused on using only sampling techniques and others were focusing on incremental learning such as bagging and boosting. In addition, based on our investigation there weren't any studies that focus on the time factor on their proposed framework. As running time consider one of the most important factors when dealing with the streaming data type. One of the most popular algorithms that have been proposed to deal with imbalanced chunk-based streaming data is OUSE (Peng Liu, Yong Wang, Lijun Cai, & Longbo Zhang, 2010). This method employees over and under-sampling techniques and deals with data as a series of blocks. Mainly this method propagates all previous minority examples from the previous chunk. While REA (Sheng Chen & Haibo He, 2011), combines all hypotheses built over time in a dynamically weighted manner to make predictions on the testing data set, it mainly focuses on IR that is static in all provided chunks. However, in terms of the Learn++ family such as learnppCDS and learnppNIE (Haibo He & Yunqian Ma, 2013), it focuses on applying the base learner of multilayer perceptron (MLP) to create multiple ensemble hypotheses on each chunk. In this paper, our approach is to find a way that minimizes the running time of the algorithm which is proposed in ICSM. The main goal of ICSM is to get the least imbalanced stream chunk between Maj and Min classes. Furthermore, avoiding the wastage of resources and minimize in the time involved to run the model.

3. Imbalanced Chunk-Based Stream Model (ICSM)

In this paper, an adaptive learning model that deals with imbalanced streaming data called the Imbalanced chunk-based Stream Model (ICSM) had been proposed. The main aim and goal of this model are to provide a flexible and effective method that will increase the recognition rate of minority class occurrences with least time and complexity required. As mentioned before that the main reason for focusing on the minority or so-called rare events is because many applications in real life such as fraud detection and rare disease detection require a model that correctly predicts it and doesn't bias toward the majority which is not the focus of these kinds of applications. ICSM model is processing through three main phases as shown in figure 1. First is to deal with the arriving streams, then dealing with data-level technique and finally proceed through ensemble-based technique. These phases are explained in detail in the following subsections:

3.1 Phase 1- Streaming Input Data Processing

The ICSM model starts to receive an input of data streams from the buffer to be processed, these streams of data will be process in a form of sequences of fixed size chunks. ((Xj, Y j), (X j+1, Y j+1), ...). Where X belong to the majority class, Y belongs to the minority class and j represents chunk at 0 index. These bit stream inputs have different characteristics which may change over time. As shown in figure 1, the input comes from real-world applications such as fraud detection, social network, ATM transaction, electricity, airline data streams and many more ("Massive Online Analysis Dataset,"). Moreover, the input of data stream can be generated or synthetic through artificial data generators which are developed especially for streaming data researches such as STRAGGER, STREAM-GENERATOR and many more (Paweł Ksieniewicz & Paweł Zyblewski, 2020; Jacob Montiel, Jesse Read, Albert Bifet, & Talel Abdessalem, 2018).

In synthetic data stream environment, the characteristics of the data stream can be determined depending on the specified scenarios such as the imbalanced degree or known as imbalanced ratio (IR), number of classes (e.g. Binary classes, Multi Classes), drift type (e.g. Sudden, Gradual), stream length which refer to the size of the data stream, and the chunk size where it can have different sizes or fixed chunk size. In the chunks processing step, at each time streams of chunks are received it will be processed and separate the data inside the chunk into two classes: Majority class and Minority class. However, if the unique value of the class label is determined and there is only one class exists in the chunk, an error will be raised "Only one class in processed data. Use bigger data chunk". Therefore, the model will drop this data and return to determine the chunk size value again in order to process binary classes as it is the main aim of this research (e.g., as an example if size of the chunk is 50 samples and the ICSM can only find one class with label "zero", the ICSM cannot work properly. Therefore, another chunk size will be specified till it have two classes at each chunk).

3.2 Phase 2 - Data-level Technique

Sampling technique is an approach used to solve the problem of imbalanced classes by using one of the three techniques or by combining more than one of them. These techniques are over-sampling, under-sampling and hybrid approach (Néstor Rodríguez, David López, Alberto Fernández, Salvador García, & Francisco Herrera, 2021). In ICSM, we use an over-sampling technique to avoid losing valuable information from the stream (Michał Koziarski, 2021). If the first chunk arrives, the imbalanced ratio will be calculated using formula.1. It will be used for each chunk once at a time. If IR is detected, which is equal or less than the specified threshold, then Synthetic Minority Oversampling (SMOTE) (Alberto Fernández, Salvador Garcia, Francisco Herrera, & Nitesh V Chawla, 2018) oversampling technique will be used. SMOTE is designed specially to tackle the imbalanced data problem to adjust the classes by synthetically generating

samples of underrepresented concepts. The judgment for these samples is based on Euclidean Distance (Tora Fahrudin, Joko Lianto Buliali, & Chastine Fatichah, 2019) between that data points inside the current chunk which starts by randomly choosing sample from processed samples in the chunk. Afterwards, it will select one of k nearest neighbors the data inside the chunk, which will be used to generate synthetic pattern. Otherwise, if the data consider to be balanced where the IR is more than or equal specified threshold then the chunk will be trained by moving to the last phase without passing through pre-process SMOTE technique.

$$\left(IR = \frac{Number\ of\ observation\ of\ majority\ Class}{Number\ of\ observation\ of\ minority\ Class}\right), \ldots . (IR)_n \qquad \textbf{(1)}$$

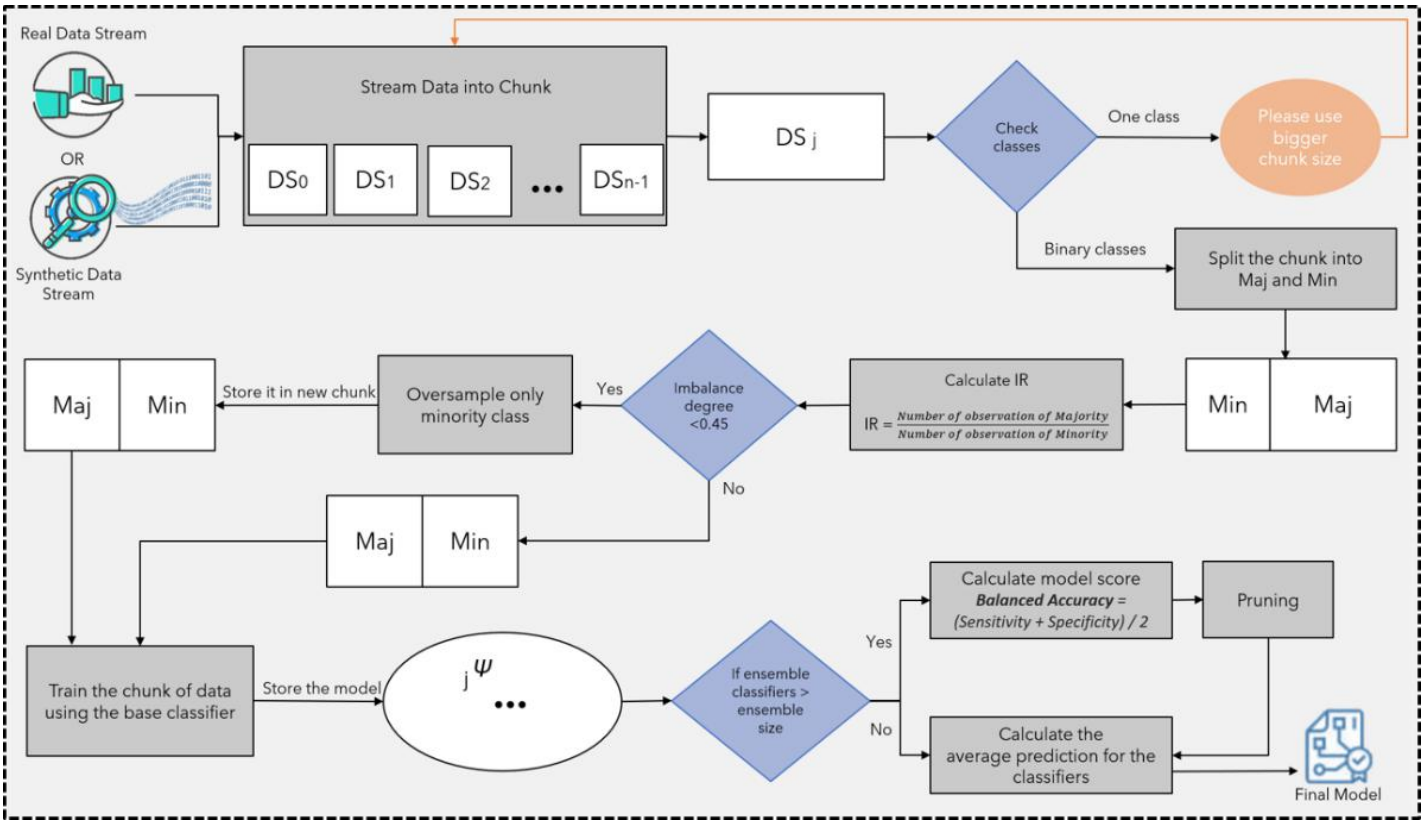


Figure 1: ICSM Framework Detailed Diagram

3.3 Phase 3 - Ensemble Classifier

The Final phase in the ICSM is the Ensemble-based classifier. ICSM combines the sampling technique that was explained in the previous section with ensemble-based classifier technique. Ensemble-based is an approach to deal with imbalanced data and non-stationary environment. It combines several models to enhance the performance of a single classifier model. ICSM follow an ensemble-based approach that uses a base classifier (e.g., KNN, GUSSAN and etc.) and combines several models that uses same classifier in order to obtain a more optimal result. The reason behind combining these models, is that most of existing stream classifiers cannot be used to classify the skewed data streams, as it processes the data as completely balanced and at the end it will biased toward the majority class and treats the minority class or the rare event as a noise. As shown in figure 1, after oversampling the minority class in the current chunk, the chunk will be trained using KNN as a base classifier. The model of the trained chunk will be store in the ensemble where the ensemble is empty at the beginning and each time a chunk is received it will be trained and stored in the ensemble till maximum size is reached.

In the ICSM the ensemble size is equal to "ES" value, if the maximum ensemble size that is equal to "ES" has not been reached yet, then a new data chunk processed to be trained using the same base classifier and simply added to the ensemble. Otherwise, the quality of the trained classifier model is first evaluated based the balanced accuracy score metric (Guo Haixiang et al., 2017). Then pruning criteria will be applied in order to remove the model with the worst performance from the ensemble. All models with balanced accuracy scores lower than a given threshold α will be removed. Afterwards, the new model replaces an existing model whose quality is worse than the quality of the new model on this training chunk. Ensemble pruning (Omar A Alzubi et al., 2020) can help in providing a smaller ensemble size which leads to reducing the complexity of the framework and the overall processing time. Moreover, it seeks for a trade-off between the classification error rate on a validation data set and the cost to evaluate classifiers. The predictions given by the ensemble are based on the majority voting technique (Florin Leon, Sabina-Adriana Floria, & Costin Bădică, 2017). This approach has been shown to recover faster in non-stationary environment than single classifier. A weighted voting method is then used where the output of the (weighted) voting method y(x) for an instance x is given by the following mathematical expression:

$$y(x) = \arg\max_{c_j} \sum_{i=1}^{k} w_i m_i(x, c_j) \quad (2)$$

Where x be the chunk and mi,i= 1..k a set of models that output a probability distribution mi(x, cj) for each class cj,j= 1..n.

The output of the ICSM framework is to produce a more balanced data streams that is evaluated based on several metrics including balanced accuracy score and G-mean score.

4. Model Evaluation

4.1 Datasets

As one of the main aims of this research is to evaluate the model performance against the diversity of the imbalanced ratio. Therefore, several data will be generated using stream - generator package. The synthetic data that has been generated have several characteristics as shown in table I, these data can be either static imbalanced ratio as shown in figure 2, which means that for each chunk same IR is generated (e.g., data that have number of chunks = 100 the first chunk IR=0.28 then the rest of the chunks till 99 will have the same value). On the other hand, these data can have dynamic imbalanced ratio as shown in figure 3, which means that IR can be changes over several chunks (e.g., data that have number of chunks=100 the first chunk can have IR=0.28 while the second one has IR=1 and etc).

Table I: Data Stream Details

Data	Majority Samples	Minority Samples	Size	Number of chunks	Weight	IR
D1	44622	5378	50,000	100	0.1:0.9	0.1
D2	39733	10267			0.2:0.7	0.28
D3	10267	29967			0.4:0.6	0.6
D4	25122	24878			0.5:0.5	1
D5	133821	16179	150,000	300	0.1:0.9	0.1
D6	104379	45621			0.2:0.7	0.28
D7	89507	60493			0.4:0.6	0.6
D8	75991	74009			0.5:0.5	1
D9	Dynamic Data stream		150,000	200	Figure 3	

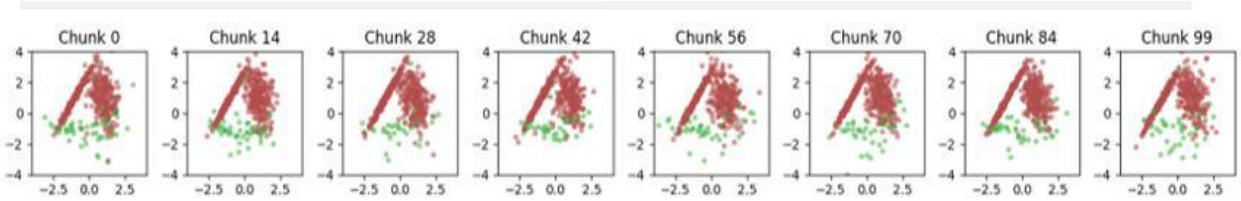


Figure 2: Static Imbalanced Data Stream Example

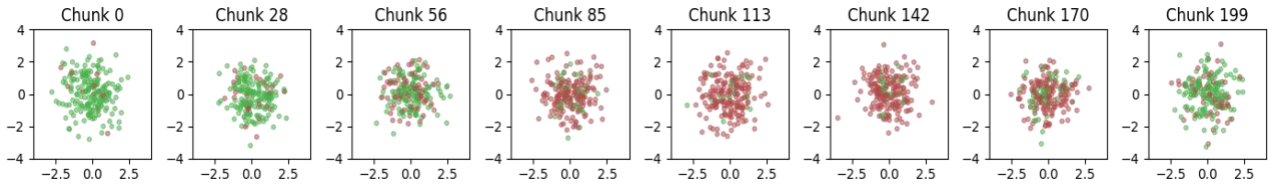


Figure 3: Dynamic Imbalanced Data Stream Example

4.2 Evaluation Measures

One of the most evaluations measures that will be used to determine the ICSM performance is the balanced accuracy score (BAS) which is a metric that is specialized in imbalanced data field (Guo Haixiang et al., 2017). The normal accuracy metric are not compatible to measure the imbalanced classes since it will ignore the probability of rare class. BAS can be calculated using sensitivity which represent the actual minority class that had been correctly identified plus specificity which represent the proportion of negative examples that has been detected divided by two as shown in formula 3.

$$\text{Balanced Accuracy Score (BAS)} = \frac{Sensitivity + Specifity}{2} \quad \textbf{(3)}$$

Another important metric is Geometric mean (G-mean) (Guo Haixiang et al., 2017), which measures the balanced performance of a learning algorithm that considers the relative balance of the classifier's performance on both the positive and the negative classes. This metric is seeking a balance between the sensitivity and the specificity as shown in formula 4.

$$G - Mean = \sqrt{Sensitivity * Specifity}$$

(4)

Finally, F1-score (Guo Haixiang et al., 2017) shows how accurate a model is by showing how many correct classifications are made. F1-score has a range between 0 and 1. The greater the score, the better the performance of the model as shown in formula 5.

$$\text{F1} = \frac{2 * TP}{2 * TP + FP + FN}$$

(5)

4.3 Experimental Setting

In this research, the ICSM will be evaluated based on different scenarios in order to deeply investigate the robustness of the model. For the framework experiments, all experiments were executed using an HP ENVYx360 Convertible 15-cn0xxx with a 1.99 GHz Intel Core i7-8550u CPU, 16.9 GB Memory (Omar A Alzubi et al.), and 64-bit operating system. The performance of the proposed model is implemented using python programming language with PyCharm editor. Pythons include number of packages specialized in streaming data type where it was inspired from MOA software and designed especially for python programmers. One of these packages are known as Stream-learn which is designed to be compatible with drifting and imbalanced data stream environment (Paweł Ksieniewicz & Paweł Zyblewski, 2020). In addition, it has stream generator for synthetic data stream and data stream evaluating techniques such as Test-Then-Train and Prequential (Paweł Ksieniewicz & Paweł Zyblewski, 2020). Moreover, another package in python for streaming data called scikit-multiflow which is intended for applications that contain continuously generated data and must be processed on the go, it also includes learning methods that exposed to new data at single scan.

In synthetic data stream several scenarios have been used to determine the ICSM performance, as in this paper we combined sampling with ensemble-based, then these are overlapping scenarios to determine the most optimal solution for the model.

Scenario 1: The diversity of imbalanced ratio, as real-world scenarios can have different situations, either highly, moderate, or low-class imbalance.

Scenario 2: The sampling technique, as ICSM's main goal is to reduce the overall running time for imbalanced stream data. Therefore, several sampling techniques will

be used. This includes the most popular oversampling techniques (SMOTE, SVM-SMOTE, Random Oversampling, Borderline-SMOTE and ADYSAN).

Scenario 3: The chunk size, as one of ICSM main goal is to propose an algorithm for chunk-based learning, the reason for choosing this factor is that chunk size can play a crucial role in the performance of the model. As too small chunk size can cause the ICSM to build improper model, beside it can cause an error since there isn't enough training data. On the other hand, using too large chunk size can cause different concepts situation without proper allocation of the data. Nevertheless, ICSM should compromises between time and number of data instances.

Scenario 4: Using several classifiers including KNN, GUSSAN, Decision Tree and MLP classifier with different IR weight in addition to different sampling techniques, in order to determine the most optimal sampling for imbalanced stream data scenario and less complex one.

Scenario 5: Compare ICSM against state-of-art. In this research we will deal with chunks of streams. So, several state-of- art algorithms will be compared with. These algorithms are proposed to be used for chunk-based scenarios as ICSM proposed algorithm. It includes Recursive Ensemble Approach (Jacob Montiel et al.), Over/UnderSampling Ensemble (OUSE),
LearnppCDS and LearnppNIE.

Scenario 6: Using very rare minority class, to study if ICSM works on rare classes.

5. Results and Discussion

5.1 Experiment 1- ICSM with Different Chunk Size

The ICSM have been tested using several chunk sizes. This includes 50,100, 500, 1000 and 3000. However, during ICSM testing phase on chunk size equal to 50 and 100 respectively, an error has been accrued as shown in figure 4.

```
    raise ValueError(
ValueError: The target 'y' needs to have more than 1 class. Got 1 class instead
```

Figure 4: Class Error

This kind of error is related to the training set in the chunk, as there wasn't enough data and only one class label exists in the processed chunk. As previously discussed, the ICSM framework will not run if the number of class labels less than two. ICSM has been tested using chunk size equal to 500. The results are as shown in table II that represents ICSM with different chunk sizes, as mentioned before that chunk sizes equal to 50 and 100 were not valid. On the other hand, when ICSM has been tested on chunk sizes equal to 500,1000 and 3000 it was valid and the performance was nearly close to each other in term of the evaluation metrics. However, in term of the running process (fig 6), the larger the chunk size the more delay the ICSM will take in seconds to run the model.

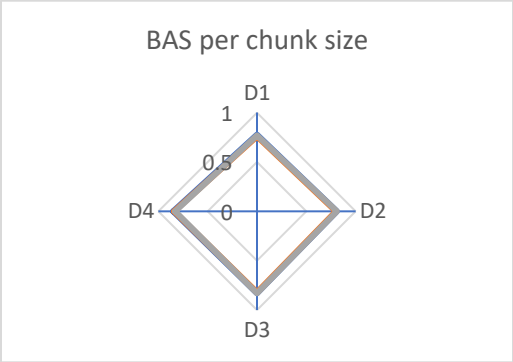


Figure 5: BAS

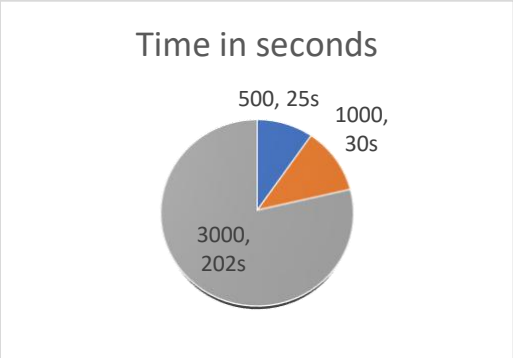


Figure 6: Time in seconds

Table II:ICSM Vs chunk size

Chunk Size	Data	BAS	F-Score	G-mean	Time Delay
50	D1 - D9	Not Valid			
100	D1 - D9				
500	D1	0.7786	0.862297	0.867681	25
	D2	0.8126	0.865386	0.868214	23
	D3	0.8314	0.874637	0.875418	23
	D4	0.850252	0.847946	0.849669	20
	D5	0.775044	0.8730896	0.868407	26
	D6	0.823182	0.8800391	0.877559	23
	D7	0.849121	0.897641	0.8961734	22
	D8	0.86326945	0.8677594	0.867135	25
	D9	0.800757	0.810179	0.819492	26
1000	D1	0.76796	0.852302	0.857233	30
	D2	0.809759	0.860534	0.863181	30
	D3	0.829662	0.876315	0.877108	29
	D4	0.842742	0.844181	0.844379	28
	D5	0.736335	0.83577	0.842781	30
	D6	0.772771	0.83108	0.835094	30
	D7	0.789129	0.846143	0.847012	30
	D8	0.808203	0.809166	0.809386	29
	D9	0.748256	0.765657538	0.775386	30
3000	D1	0.773085	0.864867505	0.86962	202
	D2	0.811374	0.852879	0.855804	201
	D3	0.831617	0.876898	0.877543	198
	D4	0.838208	0.839443	0.839534	191
	D5	0.756899	0.862254	0.866833	210
	D6	0.794259	0.848626	0.851033	212
	D7	0.814466	0.870939	0.872436	205
	D8	0.823396	0.830165	0.830807	200
	D9	0.782717	0.788226	0.797334	300

5.2 Experiment 2 – Comparing Sampling Technique Performance

The second experiment is to test the sampling technique quality on the overall performance of the ICSM framework with different IR diversity in conjunction. ICSM have been tested on five main and most popular over-sampling techniques as described precisely (SMOTE, SVMSMOTE, Random Sampling, Borderline- SMOTE and finally ADYSAN). With a base classifier similar to state-of-art techniques which is KNN as shown in figure 7,8 and 9, the results of sampling technique were nearly the same for the five of them in term of BAS, G-mean and F-score for moderate, low and balanced IR diversity. However, in term of the data that have higher imbalanced degree which weighed [0.1,0.9], Random oversampling was nearly having the lowest BAS. While in term of G-mean score as shown in figure 8 SVM-SMOTE was having the higher g-mean score in term of the four diversity of IR weight.

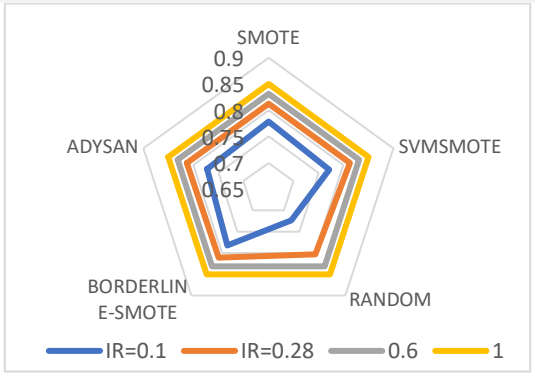


Figure 7: Balanced Accuracy Score

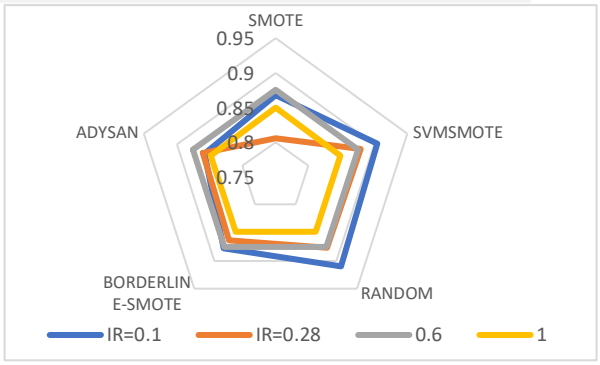


Figure 8: G-Mean Score

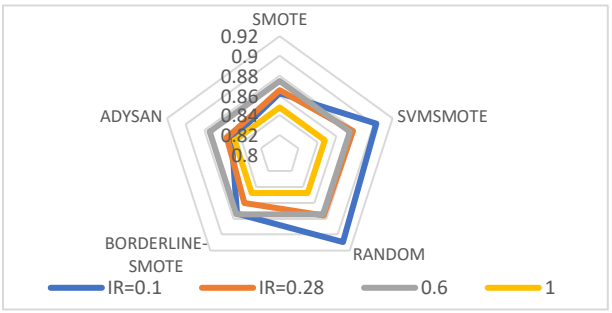


Figure 9: F-score

5.3 Experiment 3 – Comparing Base Classifier Quality on the framework Performance

We test the sampling technique SMOTE with other types of classifiers to study the effect of the different classifiers over the overall quality performance. As shown in figure 10, ICSM tested with four main classifiers (KNN, GUSSAN, Decision Tree, and MLP classifier) on two different number of chunks. The first chunk is 300 and as shown that CUSSAN is performing better than other classifiers. Moreover, in terms of the number of chunks equal to 100, the results show that KNN and MLP have the least performance compared to other classifiers in terms of the performance of BAS. While the G-Mean score for both numbers of chunks was relatively the same as shown in figure 10. GUSSAN is performing better than other classifiers in recognizing majority and minority classes afterwards Decision Tree classifiers are on the second rank. Therefore, as one of our objectives is to minimize running time in recognizing minority class, the running time is calculated for all five classifiers as shown in tables III, IV and V for both number of chunks 100 and 300 respectively, GUSSAN and Decision Tree classifiers have the least running time compared to other classifiers for all IR diversity data sets.

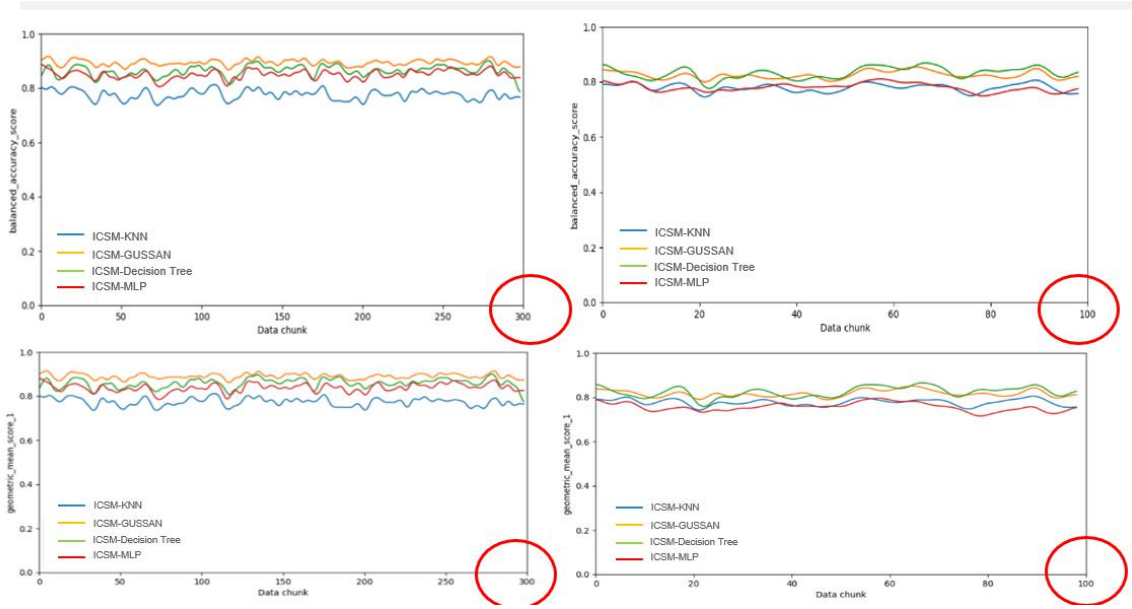


Figure 10: BAS And G-mean

Table III: Scenario2- Testing the effect of different Classifier on the same sampling technique

Table IV:

Dataset	Classifier	Sampling	BAS	G-Mean	Runtime
D1	KNN	SMOTE	0.7786	0.867681	25
	GUSSAN		0.824839	0.94415	6
	Decision Tree		0.832052	0.824218	6
	MLP-Classifier		0.779668	0.756734	190
D2	KNN		0.8126	0.805436	24
	GUSSAN		0.90775	0.94446	6
	Decision Tree		0.912183	0.911592	7
	MLP-Classifier		0.884371	0.882874	100
D3	KNN		0.8314	0.875418	30
	GUSSAN		0.916841	0.928987	7
	Decision Tree		0.926468	0.92621	7
	MLP-Classifier		0.902194	0.901812	150
D4	KNN		0.8502	0.849669	23
	GUSSAN		0.914068	0.911975	6
	Decision Tree		0.93109	0.930946	6
	MLP-Classifier		0.912822	0.912513	89

Scenario 2 – Testing the effect of several classifiers on similar sampling technique

Dataset	Classifier	Sampling	BAS	G-Mean	Runtime
D5	KNN	SMOTE	0.775628	0.873857	26
	GUSSAN		0.893704	0.970641	6
	Decision Tree		0.863178	0.858825	7
	MLP-Classifier		0.849869	0.838519	200
D6	KNN		0.823183	0.880039	25
	GUSSAN		0.931581	0.962061	7
	Decision Tree		0.912183	0.911592	7
	MLP-Classifier		0.884371	0.882874	120
D7	KNN		0.849121	0.897641	30
	GUSSAN		0.93607	0.953335	6
	Decision Tree		0.926468	0.92621	7
	MLP-Classifier		0.902194	0.901812	155
D8	KNN		0.863269	0.867759	23
	GUSSAN		0.935394	0.935006	6
	Decision Tree		0.93109	0.930946	6
	MLP-Classifier		0.912822	0.912513	92

Table V: dynamic stream data

Data set	Sampling	Classifier	BAS	G-mean	F-score	Time/s
D9	SMOTE	KNN	0.800757	0.819492	0.810179	18
		GUSSAN	0.923061	0.912799	0.911927	4
		Decision Tree	0.893056	0.882585	0.882124	4
		MLP Classifier	0.894388	0.892888	0.892542	200

5.4 Experiment 4 – Comparison with state-of-art methods

This section presents the fourth experiment which contains comparing the ICSM method with the KNN base classifier in addition to ICSM with GUSSAN classifier with state-of-art methods that were previously described in the related work section (OUSE, REA, Learn++CDS, Learn++NIE). All these methods are ensemble-based which means they must have a base classifier. As shown in Figures 11 and 12, ICSM model achieves nearly the same performance in terms of BAS and G-mean compared when compared to the four state-of-art chunk-based methods. However, when we

dealt with highly imbalanced data as shown in figure 13, ICSM-KNN and ICSM-GUSSAN are performing better in terms of BAS and G-mean in addition to improvement in running time. As LearnppNIE is consuming 2200.32 seconds in order to generate the output, LearnppCDS and REA are consuming 323.5, 251.16 seconds respectively. However, OUSE consumes 30 seconds which is better than the others. However, the BAS and G-mean have the lower performance compared to others. Moreover, when we make an experiment for dynamic imbalanced data as shown in figure 14, the results show that ICSM performs better than the state-of-art techniques in terms of BAS, G-mean, and running time.

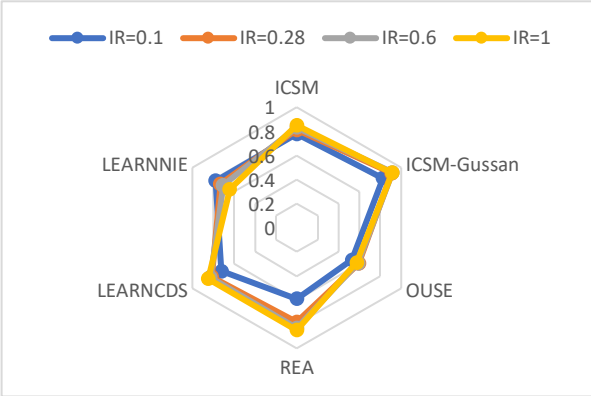


Figure 11: BAS for data stream 50,000

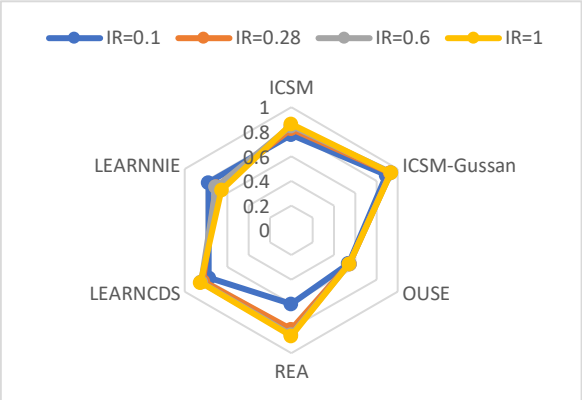


Figure 12: BAS for data stream 150,000

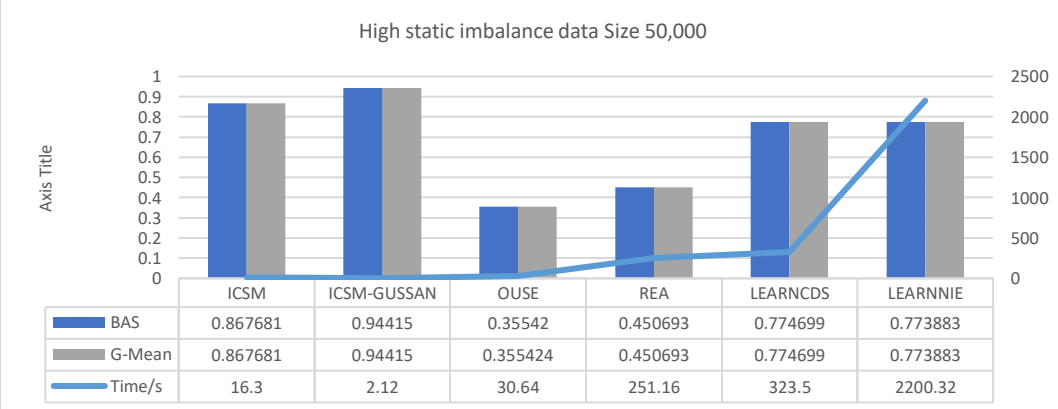
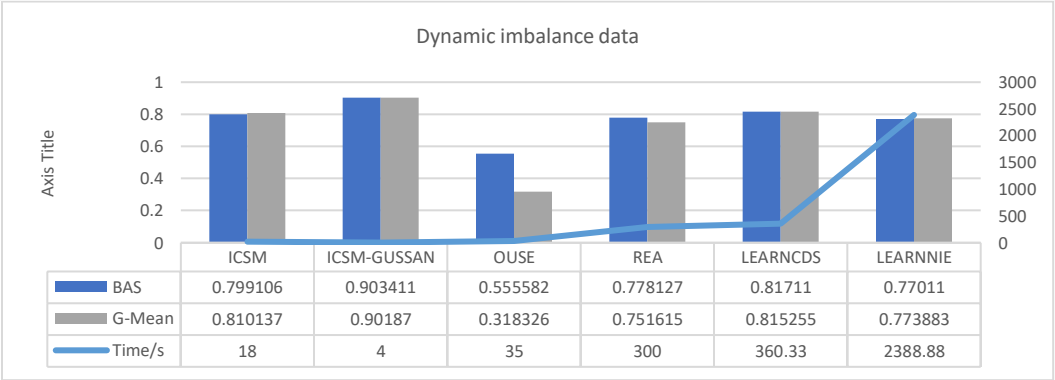


Figure 13: Highly imbalanced data comparison 50,000 Data stream



	ICSM	ICSM-GUSSAN	OUSE	REA	LEARNCDS	LEARNNIE
BAS	0.799106	0.903411	0.555582	0.778127	0.81711	0.77011
G-Mean	0.810137	0.90187	0.318326	0.751615	0.815255	0.773883
Time/s	18	4	35	300	360.33	2388.88

Figure 14: Dynamic imbalanced data stream

5.5 Experiment 5 – Rare Minority Class

The final experiment is to test the ICSM of very rare minority class and observe how ICSM and the state-of-art algorithms respond to this kind of data streams. As shown in figure 15, the red represents the majority class, and the green dots represent the minority class which is considered to be very rare comparing to other data streams that has been tested before. Based on the experiments, the ICSM with different sampling techniques and classifiers took more time delay to run comparing to the previous data streams that has been tested before. As shown in table VI, ICSM took on average 30 seconds to run the model. While other state-of-art techniques took more time delay than ICSM.

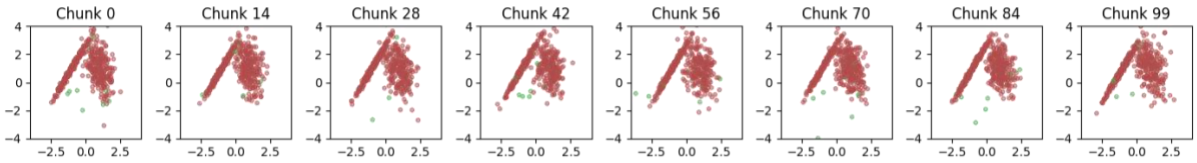


Figure 15: rare minority class

Table VII: Rare minority class performance

Technique	BAS	G-mean	Time Delay
ICSM-KNN	0.676972	0.769052	30s
ICSM – Decision Tree	0.666863	0.703021	29s
ICSM- GUSSAN	0.701981	0.801022	28s
ICSM – MLP	0.676864	0.70203	32s
OUSE	0.558692	0.388575	44s

REA	0.56102	0.286023	400s
LEARNCDS	0.676634	0.667312	450s
LEARNNIE	0.652879	0.579977	3000s

6. Discussion and Conclusion

In this paper, a model in data streams is proposed - ICSM. Basically, the model works by receiving a stream in form of fixed size chunks, and each chunk is processed in a form of two classes to identify the IR diversity. These chunks will be processed into two stages: data level which employed of the popular oversampling technique, afterwards the ensemble-based which is used by combining a number of a prediction models in order to combine them into a stronger final model. The results on the artificial dataset show that the more we have a balanced data stream the more the results are nearly the same. Therefore, ICSM outperformed existing models in detecting minority classes, especially in static high imbalanced stream data. In addition to the dynamic imbalanced data stream. Also, ICSM is performing better that the state of art methods in terms of running time.

The above analysis leads to the following conclusions. Overall, ICSM's results demonstrate a strong effect of the quality of time delay and is affected by the type of classifier. If low processing time is expected to run the ICSM, the GUSSIAN, KNN and decision tree should be chosen. When the high imbalanced ratio is expected to be dealt with, then the SVMSMOTE technique should be used instead of SMOTE. Moreover, ICSM can be used in dynamically imbalanced data streams where it achieves similar performance to the state-of-art techniques with better time delay with about 94% and more. The experiments showed that ICSM is classified with comparable quality to other state-of-art techniques and slightly outperforms them, especially when dealing with highly imbalanced data, and average time delay.

Therefore, ICSM has not been tested against the changes in the prior probabilities of concept drifts. Therefore, we cannot determine if ICSM is completely resistant to a different type of concept drift's appearance.

As for future work, several points have been considered in order to extend the ICSM model and improve the recognition rate of classes label in the streaming data field as follow:

- Improving the model by conducting more experiments on several drift types including gradual, sudden, and incremental drift.
- Maximize the data stream size up to one million.
- Extend the ICSM to work on multi-class labels.
- Enhance the model by combining it with a drift detector algorithm.

7. References

Alfhaid, M. A., & Abdullah, M. (2021). Classification of Imbalanced Data Stream: Techniques and Challenges. *Artificial Intelligence, 9*(2), 36-52.

Alzubi, O. A., Alzubi, J. A., Alweshah, M., Qiqieh, I., Al-Shami, S., & Ramachandran, M. (2020). An optimal pruning algorithm of classifier ensembles: dynamic programming approach. *Neural Computing and Applications, 32*(20), 16091-16107.

Chen, S., & He, H. (2011). Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach. *Evolving Systems, 2*(1), 35-50.

Fahrudin, T., Buliali, J. L., & Fatichah, C. (2019). Enhancing the performance of smote algorithm by using attribute weighting scheme and new selective sampling method for imbalanced data set. *Int J Innov Comput Inf Control, 15*, 423-444.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research, 61*, 863-905.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications, 73*, 220-239.

He, H., & Ma, Y. (2013). Imbalanced learning: foundations, algorithms, and applications.

Koziarski, M. (2021). *CSMOUTE: Combined synthetic oversampling and undersampling technique for imbalanced data classification.* Paper presented at the 2021 International Joint Conference on Neural Networks (IJCNN).

Ksieniewicz, P., & Zyblewski, P. (2020). stream-learn--open-source Python library for difficult data stream batch analysis. *arXiv preprint arXiv:2001.11077*.

Leon, F., Floria, S.-A., & Bădică, C. (2017). *Evaluating the effect of voting methods on ensemble-based classification.* Paper presented at the 2017 IEEE international conference on INnovations in intelligent SysTems and applications (INISTA).

Li, J., Wang, X., Lin, Y., Sinha, A., & Wellman, M. (2020). *Generating realistic stock market order streams.* Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence.

Liu, P., Wang, Y., Cai, L., & Zhang, L. (2010). *Classifying skewed data streams based on reusing data.* Paper presented at the 2010 International Conference on Computer Application and System Modeling (ICCASM 2010).

Massive Online Analysis Dataset. Retrieved Online; accessed 2-December-2021 https://moa.cms.waikato.ac.nz/datasets/

Montiel, J., Read, J., Bifet, A., & Abdessalem, T. (2018). Scikit-multiflow: A multi-output streaming framework. *The Journal of Machine Learning Research, 19*(1), 2915-2914.

Rodríguez, N., López, D., Fernández, A., García, S., & Herrera, F. (2021). SOUL: Scala Oversampling and Undersampling Library for imbalance classification. *SoftwareX, 15*, 100767.

Song, C.-W., Jung, H., & Chung, K. (2019). Development of a medical big-data mining process using topic modeling. *Cluster Computing, 22*(1), 1949-1958.

Zeadally, S., & Bello, O. (2021). Harnessing the power of Internet of Things based connectivity to improve healthcare. *Internet of Things, 14*, 100074.

8. Authors

- Mashaal.A.Alfhaid. Studying her master's degree in computer information system Department at the Faculty of Computing and Information Technology, King Abdul-Aziz University, Jeddah, Saudi Arabia. Her research field's interest includes Data Science and Machine Learning. **ORCID:** 0000-0002-5349-0248
- Manal.A.Abdullah, Prof. *received her PhD in computers and systems engineering, Faculty of Engineering, Ain-shams University, Egypt, 2002.* She has experienced in industrial computer networks and embedded systems. Her research interests include Artificial Intelligence, performance evaluation, WSN, IoT, network management, Big Data analysis, and streaming data analysis. Prof. Dr. Abdullah published more than 200 research papers in various international journals and conferences. Currently she is a **professor** in faculty of Computing and Information Technology FCIT, King Abdulaziz University KAU, Saudi Arabia SA. **ORCID**: 0000-0003-2660-6011