

Concept drift and machine learning model for detecting fraudulent transactions in streaming environment

Arati Shahapurkar, Rudragoud Patil

Department of Computer Science and Engineering, KLS Gogte Institute of Technology, Belagavi, India

Article Info

Article history:

Received Nov 16, 2022

Revised Mar 16, 2023

Accepted Apr 7, 2023

Keywords:

Class imbalance

Concept drift

Deep learning

Ensemble learning

Intrusion detection system

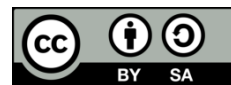
Machine learning

Social network

ABSTRACT

In a streaming environment, data is continuously generated and processed in an ongoing manner, and it is necessary to detect fraudulent transactions quickly to prevent significant financial losses. Hence, this paper proposes a machine learning-based approach for detecting fraudulent transactions in a streaming environment, with a focus on addressing concept drift. The approach utilizes the extreme gradient boosting (XGBoost) algorithm. Additionally, the approach employs four algorithms for detecting continuous stream drift. To evaluate the effectiveness of the approach, two datasets are used: a credit card dataset and a Twitter dataset containing financial fraud-related social media data. The approach is evaluated using cross-validation and the results demonstrate that it outperforms traditional machine learning models in terms of accuracy, precision, and recall, and is more robust to concept drift. The proposed approach can be utilized as a real-time fraud detection system in various industries, including finance, insurance, and e-commerce.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Arati Shahapurkar

Department of Computer Science and Engineering, KLS Gogte Institute of Technology

Belagavi, Karnataka, India

Email: asshahapurkar@git.edu

1. INTRODUCTION

The growth of the internet significantly aid different organization/fields such as social media, In recent years, credit card fraud has been increasing since there is an increase in the usage of the internet [1]. Nowadays, many people have started utilizing their credit cards for various kinds of transactions and many people fall to scammers which may lead to fraud cases [2]. Credit card fraud refers to a scammer utilizing the user's credit card number and personal identification number (PIN) or the user's stolen credit card for financial transactions from the user's account without their knowledge. Credit card scams fall under identity theft and have become increasingly common nowadays [3], [4]. There are various ways through which credit card information is usually stolen. Some examples are skimming, dumpster diving, hacking, and phishing [5]. In addition, many scammers are employing Twitter bots to convince misguided users to send money to compromised PayPal as well as Venmo accounts. The bots seem to be launched whenever a genuine user request another one for their payment details, probably obtaining such tweets through a query for terms like PayPal, Venmo, or even other providers. By stealing another user's profile photo and coming up with an identical username, they can pass themselves off as them while asking for money from the actual tweeter. Moreover, in the past few years, Twitter spam has gotten progressively worse. Twitter's massive user base and the volume of information exchanged there both contribute significantly to the rapid spread of spam [6]. Twitter and the research community have been creating several spam detection systems by utilizing various machine-learning approaches to protect users [7]. However, a recent study found that because the features of

spam tweets change over time (data imbalance and concept drift), the current machine learning-based detection methods are unable to identify spam with accuracy. Due to data imbalance and concept drift [8], [9], fraud detection and spam detection is a particularly difficult task. Additionally, fraud detection models and spam detection models differ from conventional classification in that in the initial phase, human investigators only supply a small set of supervised samples and only have time to evaluate a small number of alerts. The vast majority of transaction labels are not made available until a few days later, after customers may have detected fraud. When learning in a concept-drifting environment, it is important to carefully evaluate the delay in getting precise labels and the interaction between alerts and supervised information. Due to all these problems, there is a requirement for a model which can detect spam and fraud which has data imbalance and concept drift issues. Hence, in this paper, we propose a model which provides a solution to these problems. Furthermore, the significance of the research is: i) the continuous stream drift-identification (CSDI) employ an effective cross-validation scheme for selecting meaningful feature during the training of the predictive model and ii) the CSDI-based attack detection model achieves much better accuracy, recall, and F-measure performance in comparison with the existing ensemble-based classification model.

2. LITERATURE SURVEY

This section provides various research work done to address the concept drift issues and also the class imbalance issues which affect the overall accuracy during classification or prediction. Bayram *et al.* [10] have presented a review of various concept drift methods which have been used over the years. They discussed the various works which have been done to resolve the issue of concept drift. Further, they have discussed how machine learning (ML) can help to reduce these issues. In this paper, the researchers have also classified the various models based on their performance. They have mentioned the challenges and possible research directions when working with a concept drift issue. Mayaki and Riveill [11] have proposed a detection model for the concept drift. This model was built based on an autoregressive model known as application discovery and dependency mapping (ADDM). In [12], a method called EISStream was proposed to detect the concept drift by utilizing the ensemble and traditional ML techniques on both artificial and real-time data. Moreover, the EISStream model uses the majority voting method, allowing just the best classifier to cast a vote and decide which classifier is the best. Further, according to [13], for the detection of the data and the issue of the concept drift, they have first proposed a drift detection strategy that makes use of the principal-component-analysis (PCA) approach to analyze the constant changes in the variance of the characteristics throughout the intrusion detection data streams for the detection of the data and concept drifts. To resolve these drift issues, this method has discussed an online deep neural network (DNN) which automatically changes or adjusts the size of the hidden layers in the neural network based on the hedge weighting technique, hence, this enables the method to easily learn the intrusion and adapt to any new intrusion. Jayaratne *et al.* [14] reviewed the existing concept drift models and concluded that the existing concept drift detection techniques are dependent on the classifier and need labeled data. However, the cyber-physical system (CPS) data streams are dynamically unstructured and unlabeled. Jayaratne *et al.* [14] have proposed an unsupervised ML model which constantly detects the concept drift in the industrial cyber-physical system. Priya and Uthra [15] have proposed a model, CIDD-ADODNN, which is efficient for the detection of class imbalance and concept drift issues by employing the ADADELTA-optimizer-based deep-neural-networks (ADODNN). This model is used to preprocess the streaming data into class imbalance and concept drift, handle the class imbalance and detect the concept drift and finally classify severely imbalanced streaming data.

The ADO-based hyperparameter tuning procedure [16] is used to find the DNN model's ideal parameters to improve the classifier performance. Liu *et al.* [17] have demonstrated how the current data augmentation techniques either neglect the distribution of data or the spatial relationships among the features. To overcome the above problem, the researchers have suggested a network anomaly detection method using the convolutional neural network (CNN) which is based on data augmentation and feature representation (NADS-RA). With the aid of the least-squares generative-adversarial-network, which mitigates the impact of an unbalanced training set and prevents over-fitting. Also, an image-based augmentation method is created to create an augmented image under the distribution pattern of rare network anomaly images. Following that, NADS-RA is applied to the CNN classification model. Zhou *et al.* [18] have proposed an algorithm, correlation feature selection using the bat algorithm (CFS-BA), for intrusion detection systems which is based on ensemble and feature selection methods. The proposed algorithm chooses the best subset based on the correlation among the various features which has been presented in the initial stage for the dimensionality reduction. Yotsawat *et al.* [19], they have proposed a novel ensemble method known as the cost-sensitive neural network ensemble (CS-NNE) for creating a credit scoring model based on a cost-sensitive neural network. The multi-base neural networks can take into account unbalanced classes in the suggested method because multiple class weights are modified to the original training data. A novel ensemble architecture that

may successfully identify various attack types is proposed in [20]. The suggested method relies on ranking the detection capabilities of multiple base classifiers to recognize distinct sorts of attacks. The rank matrix for various attack categories is computed using the F1-score of an algorithm. Only the output from algorithms with the greatest F1-Score in the rank matrix for a certain attack category is taken into account for the final prediction. The voting strategy, in contrast, bases the final classification on the vote of all classifiers in the ensemble, regardless of whether the algorithm is effective enough to identify that assault or not.

3. METHOD

In this section, we discuss the standard XGBoost algorithm and its cross-validation. Further, we present a concept drift aware-machine learning framework and in the final section of the proposed methodology, a continuous stream drift identification model has been proposed. The standard XGBoost model [21], this model has an improved cross-validation technique which has been used for the selection of only the useful features. In the below section, the standard XGBoost model has been given.

3.1. Standard XGBoost model

The proposed model utilizes the standard XGBoost model which has been proposed in [21]. The standard XGBoost model in this proposed model has been used for classifying and training whether any scam or fraud is being happened during any transaction. In the standard XGBoost technique, the h_p has been used for assuring minimal loss using the greedy technique used which has been shown in (1).

$$\begin{aligned} N^p &\cong \sum_{k=1}^p \left[n(\hat{a}_k^{(p-1)} + a_k) + i_k h_k(z_k) + \frac{1}{2} j_k h_p^2(z_k) \right] + \beta(h_p) \\ &\propto \sum_{k=1}^p \left[i_k h_k(z_k) + \frac{1}{2} j_k [h_p(z_k)]^2 \right] + \beta(h_p) \end{aligned} \quad (1)$$

In the (1) the j_k has been defined for second-order gradient descent $n(\hat{a}_k^{(p-1)} + a_k)$ of and h_j has been defined for first-order gradient descent of $n(\hat{a}_k^{(p-1)} + a_k)$. Hence, the tree of the XGBoost model h_p can be attained by reducing (1). The loss function for the tree is evaluated using (2).

$$\mathcal{N}_y = -\sum_{k=1}^q (\alpha a_k \log(\hat{a}_k) + (1 - a_k) \log(1 - \hat{a}_k)) \quad (2)$$

In (2) the α is defined for the bias function which is used for describing the feature imbalance. For optimizing the feature imbalance, cross-validation is used. The cross-validation technique is constructed using various sets of K folds instead of using one set of K folds. For evaluating a single fold having cross-validation is using (3).

$$CV(\sigma) = \frac{1}{M} \sum_{k=1}^K \sum_{j \in G_{-k}} P(b_j, \hat{g}_\sigma^{-k(j)}(y_j, \sigma)) \quad (3)$$

The (3) fails to provide a good result when the data contains imbalanced values. Hence for reducing the error in the cross-validation, Shahapurkar and Rodd [21] has proposed cross-validation having two layers. These layers contain the main features of the dataset and features which have been selected using the first layer. These two layers are used for constructing a prediction model. The two-layer cross-validation technique is evaluated using (4).

$$CV(\sigma) = \frac{1}{SM} \sum_{s=1}^S \sum_{k=1}^K \sum_{j \in G_{-k}} P(b_j, \hat{g}_\sigma^{-k(j)}(y_j, \sigma)) \quad (4)$$

From the cross-validation, for optimizing the parameters and for the selection of an optimal value, (5) is used.

$$\hat{\sigma} = \arg \min_{\sigma \in \{\sigma_1, \dots, \sigma_l\}} CV_s(\sigma) \quad (5)$$

In (4), $P(\cdot)$ is used to define the gradient loss parameter, $\hat{g}_\sigma^{-k(j)}(\cdot)$ is used for the estimation of the coefficients, and the training size is defined using M . Using the standard XGBoost model with cross-validation, this technique can provide better results even when there are multiple optimization parameters.

3.2. Concept drift aware-machine learning framework

The work uses the drift detection model presented in [21] for designing an improved drift detection mechanism for a streaming environment. The framework of the proposed concept drifts aware machine learning for detecting fraudulent transactions in streaming environments is given in algorithm 1. In algorithm 1, first, the XGB classifier is trained to detect whether a given transaction is normal or malicious. Then, the work uses K-L divergence for identifying distribution between different streams (i.e., present and past data distribution) using the XGB classifier. Then, continuous stream drift-identification (CSDI) is designed to identify whether the current data stream is different from past data streams; if the drift is true, establish the drift period. Then, these streams after the drift period are used for updating the classifier. Then, the updated classifier is tested with new streams for validating the model.

Algorithm 1. Concept drift aware machine learning model

Input. Data streams $S = \{S_1, S_2, \dots, S_t\}$
 Output. Classifier C
 Step 1. Start
 Step 2. Training classifier C using S_t
 Step 3. $\forall t = 2, 3, \dots$
 Step 4. Evaluate distribution among S_t and $S_t: S_D(S_t|S_1)$ and add it to W
 Step 5. Continuous Stream Drift Identification test on W for identifying drift
 Step 6. If True
 Step 7. Retrain classifier C with streams after the drift period, $W = \emptyset$, $S_1 = S_t$
 Step 8. End If
 Step 9. Validate C on S_t
 Step 10. End \forall
 Step 11. Stop

3.3. Continuous stream drift identification model

In this section, the continuous stream drift identification model. The algorithm to identify drift on continuous stream data is given in algorithm 2. In this work obtaining a subwindow, X undersampling process is executed on a static window. After that, the work applies a continuous stream drift identification process on the subwindow and test window W using algorithm 3. Then, the present drift pointer is considered to be in static nature if the outcome is negative and is added to the static window.

Algorithm 2. Continuous stream drift identification tests

Input. Detection samples in streams sets: $S = \{S_1, S_2, \dots, S_t\}$, window size n
 Output. The drift time w^*
 Step 1. Start
 Step 2. Wait till $|S| = 2n$ considering session instance w_0
 Step 3. $\forall w = w_0, w_0 + 1, \dots$
 Step 4. Select recent n detection sample for obtaining test window W and remaining samples from S are used to obtain a static window.
 Step 5. Sampling T to obtain sub-window T
 Step 6. Execute continuous stream drift identification test on T and U on the total session stream using algorithm 3.
 Step 7. If (true) then
 Step 8. Execute continuous stream drift identification test on T and U on sub-session stream using algorithm 4.
 Step 9. If (true) then
 Step 10. Privilege the drift time w^*
 Step 11. Request to update the model after w^*
 Step 12. $E = \emptyset$, go to step 2
 Step 13. Break
 Step 14. End if
 Step 15. End if
 Step 16. Construct identification feature for S
 Step 17. End \forall
 Step 18. Stop

Nonetheless, if the outcome of algorithm 3 is positive, then test window W is further analyzed using continuous stream drift identification using algorithm 4 for localizing the drift point. If the drift point is identified, the proposed CSDI methodology is optimized with data from the drift point to the current instance. Further, when the model is expected to overflow then half of the points in S are discarded.

Every group of streams builds at least one drift indicator S_w at given session instance as:

$$S_w = 1 - \frac{1}{n_w} \sum_{j=1}^{n_w} [h^*(a_w(j) - b_w(j))]^2.$$

In particular the parameter $h^*(\cdot)$ is a pretrained model learned using the preceding static state. The parameter $a_w(j)$ defines features of j th stream at session window w and the parameter $b_w(j)$ defines the label of j th stream at session window w . Thus, $1 - \frac{1}{n_w} \sum_{j=1}^{n_w} [h^*(a_w(j) - b_w(j))]^2$ in above equation computes the mean accuracies of entire n_w data streams in a group. Once obtaining the drift identifier S and divide it into \hat{X} and W , in an iterative manner randomly select streams from \hat{X} for obtaining X until $|X| = |W|$. In general, the test window size $|W|$ provides larger significance assuring that X is representative for \hat{X} . However, at the same time because of larger size induces a higher delay for the accumulation of streams; thus, this paper introduces a sub-session-based drift identification test for establishing good drift point accuracies. This work further assumes that $H: |\mu_x - \mu_w| > 0$. There may exist certain drift points in the test window if the outcome is positive. Thus, the prerequisite is to further perform continuous stream drift identification on sub-session-based drift identification for localizing the perfect drift point. In improving computational efficiency in this work using algorithm 4 is used for dividing the test window into static and non-static ones. Let $X^{(1)}: \{X_1, X_2, \dots, X_n\}$ be self-determining streams with $X \sim N(\beta_1, \sigma^2)$. Let $W^{(2)}: \{W_1, W_2, \dots, W_{n-m}\}$ be self-determining streams with $W^{(1)} \sim N(\beta_1, \sigma^2)$, $W^{(2)}: \{W_{n-m+1}, W_{n-m+2}, \dots, W_n\}$ with $W^{(2)} \sim N(\beta_2, \sigma^2)$ and $W = W^{(1)} \cup W^{(2)}$. \bar{X} , $\bar{W}^{(1)}$, $\bar{W}^{(2)}$ and \bar{W} defines its streams average, respectively. Let $E(\cdot)$ defines the anticipated outcome of an arbitrary parameter. If $m \geq \mu n$, where $\mu = \left[1 / \left(1 + \left(\frac{w_y(n-2)}{w_y(2n-2)} \right)^2 \left[\frac{(n-4)}{n-2} \right] \right) \right]$; $E(\bar{W}^{(1)}) - E(\bar{W}^{(2)}) > 0$, then: $E(\bar{X}) - E(\bar{W}) > 0$.

Algorithm 3. Total session-based drift identification

Input. The optimized window X and test window W , the window size n

Output. If there exists drift within W

Step 1. Start

Step 2. Estimate data stream average and difference of X and W

$$\beta_X = \frac{1}{n} \sum_{j=1}^n X_j, \quad \sigma_X^2 = \frac{1}{n-1} \sum_{j=1}^n (X_j - \beta_X)^2$$

$$\beta_W = \frac{1}{n} \sum_{k=1}^n W_k, \quad \sigma_W^2 = \frac{1}{n-1} \sum_{k=1}^n (W_j - \beta_W)^2$$

Step 3. Construct two-tailed statistics

$$\sigma_{g1} = \sqrt{\frac{\sigma_X^2 + \sigma_W^2}{2}}, \quad w_1 = \frac{|\beta_X - \beta_W|}{\sigma_{g1} \sqrt{2/n}}$$

Step 4. If $w_1 \geq w_y(2n-2)$

Return 1

Else

Return 0

End If

Step 5. Stop

Algorithm 4. Sub-session-based drift identification

Input. Test window W and window size n

Output. Ideal drift point w^*

Step 1. Start

Step 2. Select recent m drift points within W for obtaining $W^{(2)}$

$$n = \mu n, \quad \mu = \frac{1}{1 + \left(\frac{w_y(n-2)}{w_y(2n-2)} \right)^2 \left[\frac{(n-4)}{n-2} \right]}$$

The other leftover point will be used to obtain $W^{(1)}$

Step 3. Estimate data stream average and difference of $W^{(1)}$ and $W^{(2)}$

$$\beta_1 = \frac{1}{n-m} \sum_{j=1}^{n-m} W_j^{(1)}, \quad \sigma_1^2 = \frac{1}{n-m-1} \sum_{j=1}^{n-m} (W_j^{(1)} - \beta_1)^2$$

$$\beta_2 = \frac{1}{m} \sum_{k=1}^m W_k^{(2)}, \quad \sigma_2^2 = \frac{1}{m-1} \sum_{k=1}^m (W_k^{(2)} - \beta_2)^2$$

Step 4. Construct two-tailed statistics

$$\sigma_{g2} = \sqrt{\frac{(n-m-1)\sigma_1^2 + (m-1)\sigma_2^2}{(n-m)+m-2}}, \quad w_1 = \frac{|\beta_1 - \beta_2|}{\sigma_{g2} \sqrt{\frac{1}{n-m} + \frac{1}{m}}}$$

```

Step 5. If  $w_1 \geq w_y(n-2)$ 
    Keep track of the time period  $w^*$  considering recent  $m$ -th point within  $W$ 
    Return 1
Else
    Return 0
End If
Step 6. Stop

```

The work assumes that the last m points in W are the ideal drift points. The streams X and W are expected to be different if there exists substantial variation between the two parts $W^{(1)}$ and $W^{(2)}$ in W . The motivating factor is using total session-based drift identification minimum quantity of drift points can be extracted; if not enough points are established then it is difficult to get a positive outcome; thus, if we can claim variance between X and W , therefore statistically it is impossible to have a significant difference between $W^{(1)}$ and $W^{(2)}$. Based on such an assumption, algorithm 4 efficiently established the ideal drift point.

4. RESULTS AND DISCUSSION

In this section, we discuss the metrics used for calculating the results. From the performance metrics, we evaluate the accuracy, precision, recall, F1-score, and false positive rate (FPR) for the credit card dataset. For comparing the results, we have used the standard machine learning models, support vector machine (SVM) [17], random forest (RF) [17], decision tree (DT) [17], network anomaly detection scheme-representation and augmentation (NADS-RA) [17], generative adversarial networks (GAN) [22], standard XGBoost, and the proposed model. Similarly, using the performance metrics, we evaluate the accuracy, recall, and F1 score. To compare the results with the existing models, we have used the standard machine learning model, RF [23], k-nearest-neighbor (KNN) [23], SVM [23], XGBoost [24], and data imbalance aware XGBoost (DIA-XGBoost) [24] and the proposed CSDI model. After this, we evaluated the drift time detection of our model. Further, discuss the data imbalance and drift problems in the credit card dataset [25] and Twitter spam dataset [26].

4.1. Performance metrics

In this section for the detection of fraud in the credit card dataset and spam in the Twitter dataset, we use the ROC curve metrics, i.e., accuracy, precision, recall, F1-score, and FPR. The following metrics are calculated as follows. For calculating the accuracy of the model, we use (6),

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (6)$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative. For calculating the recall of the model, we use (7),

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

for calculating the F1 score of the model, we use (8),

$$F1\ Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (8)$$

Using all the equations we evaluate the results which are shown below in the different sections.

4.2. Experiment on credit card dataset

In this section, the accuracy, recall, and F-measure of the proposed model compared with the existing model for the credit card fraud dataset has been evaluated. The results have been shown in Figure 1. In this section, we have evaluated the accuracy, recall, and F-measure of the ensemble model, standard XGBoost model, and the proposed model have been evaluated. The results show that the ensemble model attained an accuracy of 98.43%, the XGBoost model attained an accuracy of 98.85% and the proposed model attained an accuracy of 99.98%. The results also show that the proposed model attained a recall of 1 whereas the ensemble-based model and standard XGBoost model attained a recall of 0.6966 and 0.7977 respectively. Finally, the proposed model attained an F-measure of 0.9538, and the existing models, the ensemble-based model, and the standard XGBoost model attained an F-measure of 0.8033 and 0.8677 respectively. This shows that the proposed model can detect fraud detection more accurately when compared with the other existing models.

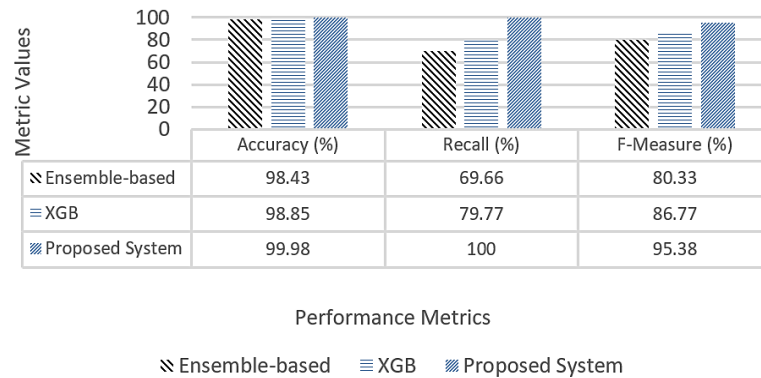


Figure 1. Performance evaluation for credit card dataset

4.3. Experiment on Twitter spam dataset

4.3.1. Performance evaluation for twitter spam dataset

In this section, the accuracy, recall, and F-measure of the proposed model compared with the existing model for the Twitter spam dataset has been evaluated. The accuracy of the existing model and the proposed model has been shown in Figure 2. The results show that the existing models, SVM, KNN, RF, Standard XGBoost, DIA-XGB, and ensemble models attained an accuracy of 65.59%, 96.36%, 98.03%, 98.19%, 99.68%, and 99.96% respectively. The proposed model attained an accuracy of 99.98% which is higher than the existing model when compared to other existing models. The recall of the existing model and the proposed model has been shown in Figure 2. The results show that the existing models, KNN, RF, XGBoost, SVM, DIA-XGB, and ensemble models attained recall of 0.51, 0.63, 0.69, 0.87, 0.93, and 0.9785 respectively. The proposed model attained a recall of 1 which is much higher when compared with the existing models. The F1 score of the existing model and the proposed model is shown in Figure 2. The results show that the existing model SVM, KNN, RF, XGBoost, ensemble, and DIA-XGB attained an F1-score of 0.2, 0.653, 0.8033, 0.8677, 0.9008, and 0.9677 respectively. The proposed model attained an F1 score of 0.9538.

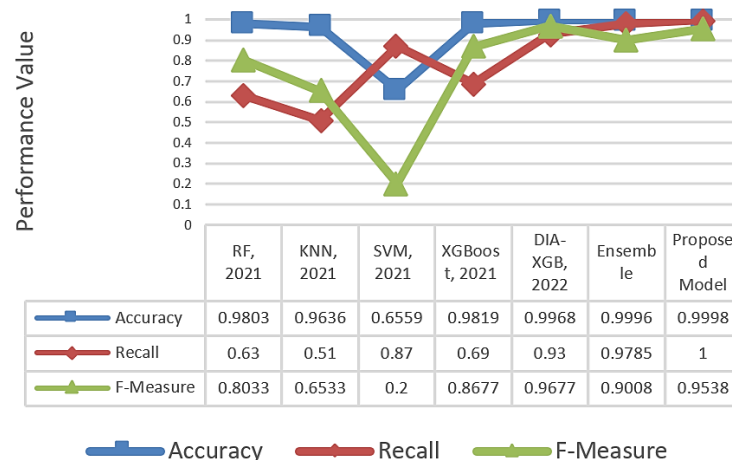


Figure 2. Performance evaluation for Twitter spam dataset

4.3.2. Drift time

In this section, the drift time of the dataset for the Twitter spam dataset has been explained. The accuracy, recall, and F-measure has been shown in Figures 3, 4, and 5 respectively. The results have been compared with the MDDT+XGB [24] model which mainly focuses on the drift. This model also handles the drift using an ensemble model which also uses the XGBoost model. The accuracy performance of the MDDT+XGB keeps fluctuating each day as seen in Figure 3. The proposed model's accuracy remains constant and keeps on decreasing as the no of days increases. There is a drop after the 5th day as the drift increases in the data. In Figure 4, the recall performance of 5 days performance can be seen. Similar to the

accuracy the recall also keeps fluctuating each day for the recall. The proposed model is constant and decreases as the number of days increases. In Figure 5, the F-measure can be seen where the existing model increases on the 2nd day but drop after the 3rd day. The F-measure of the proposed model remains constant throughout the 5 days. Hence, the proposed model is better than the existing model in handling the drift.

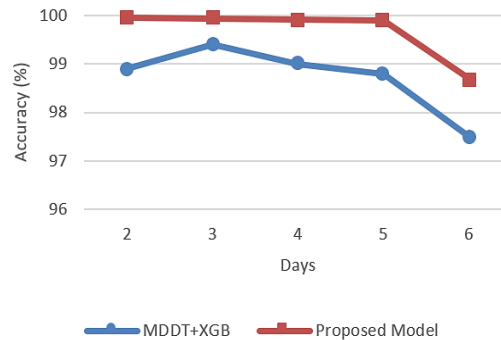


Figure 3. Accuracy performance of 5 days

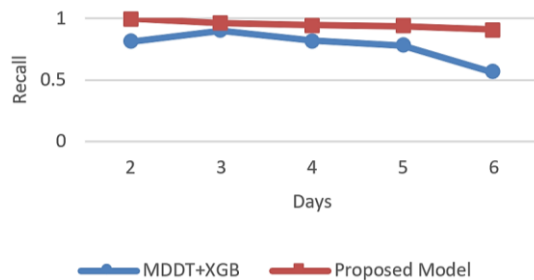


Figure 4. Recall performance of 5 days

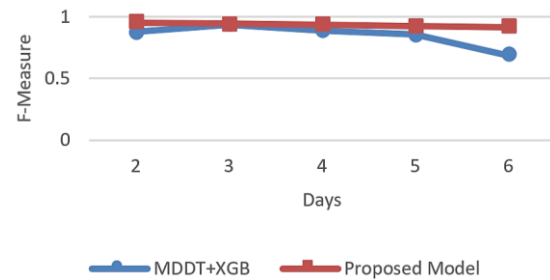


Figure 5. F-measure of 5 days

5. CONCLUSION

In this paper, first, we have studied how class imbalance and concept drift problems arise in data and how it affects the overall system during classification and detection. Further, we have studied the various models which have been presented to detect class imbalance and concept drift. After this, we have presented a model which addresses the data imbalance and drifts problems during the detection. Furthermore, we have experimented with our model with the credit card fraud dataset and Twitter spam dataset. The results show that the proposed model attains higher accuracy when compared with the existing systems for both datasets. The proposed model provides an opportunity for addressing the class imbalance and drift issues in a given dataset and predicts better when compared with the existing models. The current proposed model can detect better with credit card fraud and Twitter spam datasets. For future work, we would try to improve the model to attain higher performance metrics with other datasets.




REFERENCES

- [1] B. Bayram, B. Koroglu, and M. Gonen, "Improving fraud detection and concept drift adaptation in credit card transactions using incremental gradient boosting trees," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2020, pp. 545–550, doi: 10.1109/ICMLA51294.2020.00091.
- [2] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection and concept-drift adaptation with delayed supervised information," in *2015 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2015, pp. 1–8, doi: 10.1109/IJCNN.2015.7280527.
- [3] A. Somasundaram and S. Reddy, "Parallel and incremental credit card fraud detection model to handle concept drift and data imbalance," *Neural Computing and Applications*, vol. 31, no. S1, pp. 3–14, Jan. 2019, doi: 10.1007/s00521-018-3633-8.
- [4] A. Yeşilkanat, B. Bayram, B. Köroğlu, and S. Arslan, "An adaptive approach on credit card fraud detection using transaction aggregation and word embeddings," in *IFIP Advances in Information and Communication Technology*, Springer International Publishing, 2020, pp. 3–14.
- [5] N. Shirodkar, P. Mandrekar, R. S. Mandrekar, R. Sakhalkar, K. M. Chaman Kumar, and S. Aswale, "Credit card fraud detection techniques-a survey," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Feb. 2020, pp. 1–7, doi: 10.1109/ic-ETITE47903.2020.112.




- [6] W. Daffa, O. Bamasag, and A. AlMansour, "A survey on spam URLs detection in twitter," in *2018 1st International Conference on Computer Applications and Information Security (ICCAIS)*, Apr. 2018, pp. 1–6, doi: 10.1109/CAIS.2018.8441975.
- [7] K. s Swarnalatha *et al.*, "Spam detection in Twitter data," in *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, Dec. 2021, pp. 1–3, doi: 10.1109/CSITSS54238.2021.9683387.
- [8] A. Bechini, A. Bondielli, P. Ducange, F. Marcelloni, and A. Renda, "Addressing event-driven concept drift in twitter stream: a stance detection application," *IEEE Access*, vol. 9, pp. 77758–77770, 2021, doi: 10.1109/ACCESS.2021.3083578.
- [9] A. Priyadarshini, S. Mishra, D. P. Mishra, S. R. Salkuti, and R. Mohanty, "Fraudulent credit card transaction detection using soft computing techniques," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 23, no. 3, pp. 1634–1642, Sep. 2021, doi: 10.11591/ijeecs.v23.i3.pp1634-1642.
- [10] F. Bayram, B. S. Ahmed, and A. Kassler, "From concept drift to model degradation: An overview on performance-aware drift detectors," *Knowledge-Based Systems*, vol. 245, Jun. 2022, doi: 10.1016/j.knosys.2022.108632.
- [11] M. Z. A. Mayaki and M. Riveill, "Autoregressive based drift detection method," in *2022 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2022, pp. 1–8, doi: 10.1109/IJCNN55064.2022.9892066.
- [12] A. Abbasi, A. R. Javed, C. Chakraborty, J. Nebhen, W. Zehra, and Z. Jalil, "ElStream: an ensemble learning approach for concept drift detection in dynamic social big data stream learning," *IEEE Access*, vol. 9, pp. 66408–66419, 2021, doi: 10.1109/ACCESS.2021.3076264.
- [13] O. Abdel Wahab, "Intrusion detection in the IoT under data and concept drifts: online deep learning approach," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19706–19716, Oct. 2022, doi: 10.1109/IIOT.2022.3167005.
- [14] D. Jayaratne, D. De Silva, D. Alahakoon, and X. Yu, "Continuous detection of concept drift in industrial cyber-physical systems using closed loop incremental machine learning," *Discover Artificial Intelligence*, vol. 1, no. 1, 2021, doi: 10.1007/s44163-021-00007-z.
- [15] S. Priya and R. A. Uthra, "Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data," *Complex and Intelligent Systems*, Jul. 2021, doi: 10.1007/s40747-021-00456-0.
- [16] M. Straat, F. Abadi, Z. Kan, C. Göpfert, B. Hammer, and M. Biehl, "Supervised learning in the presence of concept drift: a modelling framework," *Neural Computing and Applications*, vol. 34, no. 1, pp. 101–118, Jan. 2022, doi: 10.1007/s00521-021-06035-1.
- [17] X. Liu *et al.*, "NADS-RA: network anomaly detection scheme based on feature representation and data augmentation," *IEEE Access*, vol. 8, pp. 214781–214800, 2020, doi: 10.1109/ACCESS.2020.3040510.
- [18] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, Jun. 2020, doi: 10.1016/j.comnet.2020.107247.
- [19] W. Yotsawat, P. Wattuya, and A. Srivihok, "A novel method for credit scoring based on cost-sensitive neural network ensemble," *IEEE Access*, vol. 9, pp. 78521–78537, 2021, doi: 10.1109/ACCESS.2021.3083490.
- [20] S. Seth, K. K. Chahal, and G. Singh, "A novel ensemble framework for an intelligent intrusion detection system," *IEEE Access*, vol. 9, pp. 138451–138467, 2021, doi: 10.1109/ACCESS.2021.3116219.
- [21] A. Shahapurkar and S. F. Rodd, "Efficient feature aware machine learning model for detecting fraudulent transaction in streaming environment," *International Journal on Information Technologies and Security*, vol. 14, no. 3, 2022.
- [22] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, vol. 479, pp. 448–455, 2019, doi: 10.1016/j.ins.2017.12.030.
- [23] X. Wang, Q. Kang, J. An, and M. Zhou, "Drifted twitter spam classification using multiscale detection test on K-L divergence," *IEEE Access*, vol. 7, pp. 108384–108394, 2019, doi: 10.1109/ACCESS.2019.2932018.
- [24] S. S. Patil and H. A. Dinesha, "URL redirection attack mitigation in social communication platform using data imbalance aware machine learning algorithm," *Indian Journal of Science and Technology*, vol. 15, no. 11, pp. 481–488, 2022, doi: 10.17485/IJST/v15i11.1813.
- [25] Machine Learning Group - ULB, "Credit card fraud detection," *Kaggle*. Accessed: June 1, 2022. [Online]. Available: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>.
- [26] NSCLab, "Twitter spam dataset," *NSCLab*, 2014. Accessed: June 1, 2022. [Online]. Available: <http://nslab.org/nslab/resources/>.

BIOGRAPHIES OF AUTHORS



Arati Shahapurkar    currently working as an Assistant Professor, at Department of Computer Science Engineering, Karnatak Law Society's (KLS) Gogte Institute of Technology, Belagavi. She has 16 years of Teaching Experience at KLS Gogte Institute of Technology, Karnataka. She has published over 10 papers in International Journals, and Conferences of High Repute. Her subjects of interest include machine learning, big data management and network security. She can be contacted at email: asshahapurkar@git.edu.



Rudragoud Patil    currently working as an Associate Professor, at Department of CSE, KLS Gogte Institute of Technology, Belagavi. He has 12 years of Teaching Experience at professional institutes across Karnataka. He published over 13 papers in International Journals, Book Chapters, and Conferences of High Repute. His subjects of interest include cloud computing, distributed computing, machine learning, and network security. He can be contacted at email: rspatil@git.edu.

Copyright of International Journal of Electrical & Computer Engineering (2088-8708) is the property of Institute of Advanced Engineering & Science and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.