International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I6P232

Volume 70 Issue 6, 308-316, June 2022
© 2022 Seventh Sense Research Group®

Original Article

Modified Weight Optimized XG Boost (MWO-XGB) for Concept Drift and Data Imbalance Problems in the Online Environment

Sagargouda S Patil[1], Dinesha H.A.[2]

[1,2]*Computer Science, Nagarjuna College of Engineering and Technology, Bangalore, Karnataka, India*.

[1]sagar.cs.kle@gmail.com

Received: 15 April 2022 Revised: 30 May 2022 Accepted: 04 June 2022 Published: 29 June 2022

Abstract – *Nowadays, many websites on the internet are being used for sharing information, connecting people, video streaming, browsing, etc. All these websites are accessed using the links which the host provides. The host provides the links with proper security and good content. But some of the sites have Malicious Uniform Resource Allocators (URL) using which the attacker can access the user information. When the user clicks or taps on the links or hyperlinks of these websites, then he is redirected to another website. In this case, the user has no idea that he is getting attacked by the user, and they are providing personal information to the attacker. Hence, in this paper, the machine learning system, XGBoost, using which the model can identify the malicious links, classify them and remove them using the proposed modified XGBoost model. In this paper, the proposed modified XGBoost method, Modified Weight Optimized XGBoost (MWO-XGB), detects the URL in an online environment with class imbalance and concept drift problems. This paper mainly focused on the popular NSL-KDD dataset and other social media datasets to identify and detect the malicious URL using the proposed model. The experimental results are better when compared with the existing system such as XGBoost etc. This model's main focus is to reduce the malicious attacks in the online environment using the MWO-XGB model.*

Keywords – *Malicious URL, MWO-XGBoost, NSL-KDD, Attack.*

1. Introduction

Today, most people can barely picture their lives without technological advancements. Furthermore, a wide range of applications, websites, and new online social websites allow many people to exchange their knowledge and build proper social and professional contact. These websites focus heavily on connecting people through sharing common interests. However, spreading information and establishing contacts with people raises some important security issues. Furthermore, a phishing/malicious Uniform Resource Locator (URL) is a URL that has been designed to be used for fraud or spam attacks. In this attack, a virus is sent by the attacker and is installed on a computer if a given user or a customer taps on a URL that contains malware. Phishing and spam are two common outcomes of malicious Uniform Resource Locators. The credentials of users are compromised when they fall victim to phishing. As a result, it is critical to distinguish between legitimate and harmful linkages. Malicious Unified Resource Locators are being used as a vector for cyber-attacks. Moreover, the phishers are constantly tweaking their cyber-attack methods. The attacker can misuse shared data for their ends.

One of the most popular forms of cybercrime is the use of a malicious website or a malicious uniform resource locator. When you're not careful, you could become a victim of fraud, including cash losses, personal information disclosures, malware installation, spyware installation, extortion, a false shopping site, an unexpected award, and so on. Visits to these sites may be prompted by email, adverts, web searches, or hyperlinks from other websites. In each scenario, the need to tap on the malicious link. The rise in phishing, spamming, and malware demand a dependable solution that categorizes and detects bad URLs. Malicious uniform resource locators are still a major source of security breaches. Malware, phishing, and spam are all frequent methods of spreading them. Black-listing is a widely used method of detecting harmful URLs. Blocklists keep track of URLs that have previously been associated with the dangerous activity. When detecting newly produced malicious URLs, these lists fall short. Machine learning algorithms have been trained. As a result, to detect dangerous URLs. In this paper, we have used the machine learning system, XGBoost, using which we can identify malicious links and classify them. In this paper, the machine learning system, XGBoost, using which the model can identify the malicious links, classify them and remove them using the proposed modified XGBoost model. In this paper, the proposed modified XGBoost method, Modified Weight Optimized XGBoost (MWO-XGB), detects the URL in an online environment with class imbalance and concept drift problems. This paper mainly focused on the popular NSL-KDD dataset and other social media datasets to identify and detect the malicious URL using the proposed model.

2. Literature Survey

This section has surveyed various machine learning methods, class imbalance, data imbalance, and drift problems in the existing systems. This paper has also surveyed the various solutions provided by various researchers to solve the problem of malicious links (URL). In [1], they have given a machine learning-based intrusion detection system that uses the Adversarial-Machine-Learning to detect the various attacks. In this method, they have used the Jacobian-based Saliency-Map-Attack. They have also analyzed how many samples can be used during the training, which will help the model work robustly. They have compared their result with the existing Random Forest and J48 methods. In [2], they have used the Software-Defined-Network and Network-Function-Virtualization in their framework to identify the various attacks. This model uses machine learning, artificial intelligence, and intrusion detection system integrated into the IoT systems to detect the attack. They have experimented with their model by building a one-class Support Vector Machine model. The results show better outcomes when compared with the existing models. In [3], various machine learning algorithms are used in the professional and academics for various purposes. They have also explained the advantages of the algorithm in the field of cybersecurity. In [4], this paper provides four semi-supervised methods for classifying spam based on hotel reviews. They have concluded that the Naïve Bayes model attains good outcomes during the self-training compared to the existing systems. This model can only be used for small datasets as training the large datasets is not efficient. In [5], they have given an intrusion detection system based on ensemble learning methods and feature selection. In this model, they have combined various machine learning algorithms, Random Forest, Forest PA, C4.5, and given an algorithm CFS-BA. They have used a voting method that takes the probability distribution of the attacker and detects the attack. They have used the standard datasets, CIC-IDS2017, AWID, and NSL-KDD, for the evaluation of the performance of their model with the existing models. In [6], they have proposed a model for the online nonstationary environments using an ensemble method. This method provides good accuracy and diversity for detection in online environments. They evaluated the model's performance using real-world and artificial datasets such as OAUE, AFWE, and DDD. In [7], they have given a model for detecting the abnormality in the network using the data augmentation, NADS-RA. They first represent the data in the image and analyze and compare the given image by rotating the original image to the left. After this, they have used the Least-Square-Generative Adversarial Network to handle the imbalance problem of the dataset. Finally, they have used the Convolutional-Neural-Network Algorithm to provide a good detection model. To evaluate the performance of their model, they have used the standard datasets, UNSW-NB15 and NSL-KDD. In [8], they have proposed a model for the detection of spam in the online social networking network, Twitter. This model uses machine learning algorithms to distinguish between spam and non-spam accounts and content. They have used a Genetic Algorithm to examine various features such as user accounts, which is further used to train the model. The model attains better results when compared with the existing models. In [9], they have proposed a cost-efficient approach to various filtrate spam. In this model, they have used a deep neural network and multi-objective-evolutionary feature selection to reduce the model's cost and filter the spam. They have evaluated their model using the social-networking spam-filtering datasets, and the results show that the model has better results when compared with the machine learning models like Naïve Bayes, Random Forest, and SVM. In [10], they have surveyed various models used to detect spam on the online social network Twitter. They also have classified spam based on irrelevant content, URL, fake users, and trending topics. They also gave some techniques based on structure, graph, content, time, and useful features.

In [11], they addressed the spam drift problem using the unsupervised machine-learning models. This model takes the input using the unlabeled tweets and checks the volatility of the spam tweets. The results show good accuracy and recall. In [12], they have surveyed the existing methods which are being used for the detection of spam on Twitter. This study shows that most of the methods use machine learning algorithms. They have also analyzed the main features the existing methods focus on, like user analysis, network analysis, content analysis, tweet analysis, etc. In [13], they have used the K-L-divergence method to identify spam. They have also used the multiscale-drift-detection-test to limit the drift problem. The results show better accuracy, f-measure, and recall performance. In [14], they have given a multiscale-drift-detection-test to restrict the drift problems in the online environment and provide a good detection model. This model has a better recall score in localizing the drift points and handles the drift problem efficiently. In [15], they proposed a framework, LightGBM, to detect irregularity in the IoT streaming environments. This model addresses the concept drift problem and gives a solution using an Optimized-Adaptive and Sliding-Windowing. In [16], they have addressed both the concept drift and data drift in IoT. They have used a machine learning-based intrusion-detection model and deep-neural network to address this problem. In [17], they have implemented and tested various concept-drift detection models to analyze how the different models perform in a scattered environment. They have also given the challenges of the concept drift problem. [18] has given an ensemble learning method that addresses the data imbalance and concept drift problem in condition-based maintenance. They have used the Linear Four Rates model to cope with the concept drift and data imbalance problem.

The attained results show that this model detects the drifts efficiently. In [19], they have addressed the problem of Spam Drift on Twitter using the Lfun method. This method can identify the spam tweets from the unlabeled tweets and then incorporates them into the classifier training process. In [20], they have presented a method, CONFRONT, which detects the problem of concept drift in the online botnets. This model classifies and optimizes the botnets and provides Denial of Service (DoS). In [21], they have proposed a method RACE to handle the concept drift problems with fewer computational overheads. This model uses the ensemble learning method to train and test the model. In [22], they have proposed a Generative Adversarial Network (GAN) model, which generates malware samples having concept drift in the given malware for the training of their model. This model also detects the concept drift and malware simultaneously and provides better results when compared with the existing models. In [23], they have addressed the problem of concept drift in imbalanced data streams. This model can detect various classes at the same time and can change the streaming environment conditions.

From all the given surveys, only some work has been carried out on the data imbalance. Moreover, there are only a few models which have presented the concept drift and data imbalance problems at the same time. Some machine learning algorithms have handled the concept drift problem and given a good result. Still, they have failed to perform well and handle the data imbalance problems in the online environment. Hence, proposed a model using the machine learning system, XGBoost, to address the data imbalance problem in the online environment and remove the malicious link from the online environment. The Modified XGBoost algorithm can simultaneously handle the concept drift and data imbalance problems in the online environment.

3. Modified Weight Optimized XGBoost (MWO-XGB)

This paper presents a Modified Weight Optimized XGBoost model, which handles class and data imbalance problems using our modified XGBoost algorithm. The architecture of the XGBoost model handling the concept drift and data imbalance is given in Figure 1. This section presents the XGBoost model, then how the XGBoosst model can handle the concept drift problems and data imbalance. Finally, present the Modified Weight Optimized XGBoost model, which attempts to remove the malicious link and check its performance using a given standard dataset.

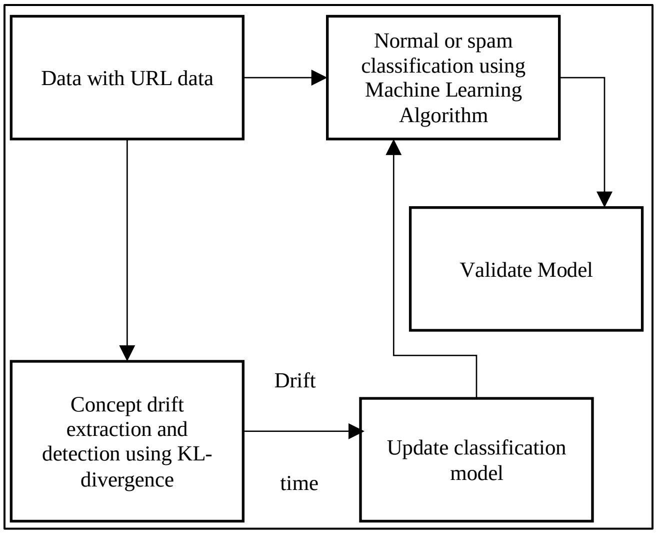
Data with URL data
Normal or spam classification using Machine Learning Algorithm
Validate Model
Concept drift extraction and detection using KL-divergence
Drift
time
Update classification model

Fig. 1 Architecture of class imbalance and concept drift aware spam drift aware classification.

An approach known as XGBoost is the best machine learning gradient tree-boosting methodology employed by many standard models to solve classification issues. Gradient tree-boosting methods are designed to combine the results of many tree classifiers. Hence, to classify the malicious link, the XGBoost model is used. In this model, the dataset having n samples having different classifications is trained using the following equation

$$\hat{Z}_j = G(Y_j) = \sum_{l=1}^{L} g_l(Y_j), \qquad g_l \in \alpha \qquad (1)$$

In Equation (1), the multi-label classification model outcomes are defined by $\hat{Z}_j$, which shows how likely a given malicious link will be categorized as belonging to a particular class based on its label. L describes the size of the tree which is used for the classification of the malicious link and l^{th} describes the likelihood that each malicious link will be classified as belonging to a certain class using the l^{th} dimension as explained below.

$$\alpha = \{g(y) = x_{t(y)}\} \qquad (2)$$

In Equation (2), all trees $g(y)$ Agree on leaf weight x and structure variable t. Minimizing the loss parameter is one of the main goals of the XGBoost classification model

$$M(G) = \sum_j m(\hat{z}_j, z_j) + \sum_l \beta(g_l) \qquad (3)$$

In the Equation (3), $m(\hat{z}_j, z_j)$ Specifies the loss function between actual and categorized outcomes.

$$\beta(g_l) = \delta U + \mu \|x\|^2 \tag{4}$$

There are three parameters in Equation (4): a penalizing term (g_l), a tree's leaf size (U), and a μ parameter governed how complex computations are. When the weighted loss function is applied to training data x, the negative log probabilistic loss function is generated by applying the following equation.

$$m(\hat{z}_j, z_j) = -\sum_k z(k) \log \hat{z}\,(m) = -\log \hat{z}\,(m) \tag{5}$$

In Equation (5), the $z(k)$ represents the k^{th} dimension of z. Also, $\hat{z}(m)$ depicts the k^{th} dimension of the output $\hat{z}$. Loss functions are optimized iteratively for the lowest possible losses, as well. The following equation can describe the optimal loss function for an iteration of u.

$$M^j = \sum_{j=1}^{o} m\left(\hat{z}_j^{(u-1)} + g_u(y_j), z_j\right) + \beta(g_u) \tag{6}$$

With the following equation, the proposed methodology establishes g_u It is a way to reduce losses greedily.

$$M^u \cong \sum_{j=1}^{o} \left[m\left(\hat{z}_j^{(U-1)} + z_j\right) + h_j g_j(y_j) + \frac{1}{2} i_j g_u^2(y_j) \right] + \beta(g_u) \tag{7}$$

The tree g_u can be found by minimizing Equation (7), where h_j represents the first-order gradient of $m\left(\hat{z}_j^{(U-1)} + z_j\right)$ and i_j represents the second-order gradient of $m\left(\hat{z}_j^{(U-1)} + z_j\right)$.

The XGBoost method can classify the malicious links but fails to handle the data imbalance problem. Moreover, the online environments contain class imbalance problems. Hence, a weight function using a heuristic-inverse function has been given to solve the class imbalance problems in the online environment and classify the malicious links. In this function, as shown in the equation below, the weight of each class is inversely proportional to the total data included in each class.

$$x_m = \frac{\overline{N}}{o_m} \tag{8}$$

In Equation (8), x_m represents the weight of the given class m, o_m represents the size of the data and $\overline{N}$ represents the average size of the data, which is given using the below equation

$$\overline{N} = \frac{\sum_m o_m}{O} \tag{9}$$

In Equation (9), the O represents the class size of the malicious links. From the Equation (8) and Equation (9), the loss function in Equation (5) can make better using the following equation

$$m(\hat{z}_j, z_j) = -\,x_m \log \hat{z}\,(m) \tag{10}$$

By increasing the weights, the model can suffer from overfitting problems. Hence, this study introduces a smooth method for calculating the weight of a specific class as outlined in the following equation to avoid the overfitting problem.

$$x_k = \frac{1}{2} + \omega * \frac{1}{1 + \exp(w_k)} \tag{11}$$

$$w_k = \frac{o_k - \overline{N}}{V} \tag{12}$$

In Equation (12), V represents the standard variance of the data size o of each class, and in Equation (11) ω is used as a controlling parameter to optimize the weight in the sigmoid function. The class imbalance problem in the online environment is handled using the above equation. Furthermore, to address the data imbalance

Further, this work addresses the concept drift problem; because the spam data varies over time. The spam drift aware classification model is described in Fig. 2. First, the Improved XGBoost classifier is trained on tweets to decide whether a tweet is normal or spam. Subsequently, using KL divergence, distributional distances between varied tweets are estimated for concept extraction. It is done to establish variation among current tweet distribution and historical tweets and carryout classifications in an adaptive manner. Then, drift detection is done to validate present tweets concepts vary concerning historical tweets and, if so, keep the drift time. Further, the drifted tweets are trained to update the classification model to improve their robustness. Lastly, the data are input to the classification model to validate its performance.

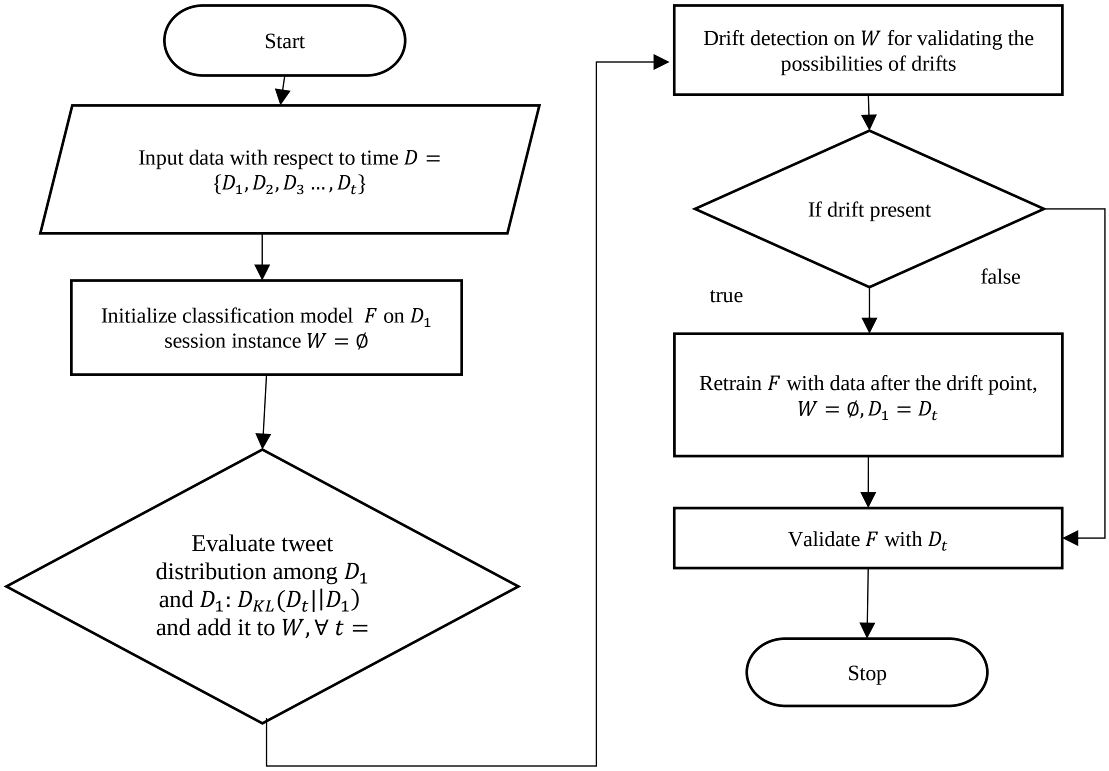
Start
Input data with respect to time $D = \{D_1, D_2, D_3 \ldots, D_t\}$
Initialize classification model F on D_1 session instance $W = \emptyset$
Evaluate tweet distribution among D_1 and D_1: $D_{KL}(D_t||D_1)$ and add it to $W, \forall\ t =$
Drift detection on W for validating the possibilities of drifts
If drift present
true
false
Retrain F with data after the drift point, $W = \emptyset, D_1 = D_t$
Validate F with D_t
Stop

Fig. 2 Flow diagram of class imbalance and concept drift aware Twitter spam classification model.

```
Algorithm 1. Drift point detection
Input. time window $W$
Output: whether any drift point present in $W$
Step 1. Split $W$ into static window $T$, test window $U$, window size $o$, and constraint coefficient $d$
Step 2. Estimate the mean and variance of $T$ and $U$
        $\beta_T = \frac{1}{o}\sum_{j=1}^{o} T_j, \quad T_T^2 = \frac{1}{o-1}\sum_{j=1}^{o}\left(T_j - \beta_T\right)^2$
        $\beta_U = \frac{1}{o}\sum_{j=1}^{o} T_j, \quad T_U^2 = \frac{1}{o-1}\sum_{j=1}^{o}\left(T_j - \beta_U\right)^2$
Step 3. Select threshold $\alpha = d\beta_T$. The threshold is optimized dynamically
Step 4. Construct two-tailed test statistics:
        $T_X^2 = \frac{T_T^2 + T_U^2}{2}, \ u = \frac{|\beta_T - \beta_U| - \alpha}{T_X\sqrt{\frac{2}{o}}}$
Step 5. If $u \leq u_\delta(2o - 2)$
            Obtain False
        Else
            Obtain True
        End if
```

Concept extraction is done for obtaining tweet information distribution. When the model establishes any drifts, it optimizes the model, ensuring the drift detection methodologies can effectively establish the outlier within. In this work, KL divergence is used for measuring similarities among two distributions using the following equation

$$D_{KL}(P\|H) = -\sum_{i=1}^{K} P_i ln\frac{H_i}{P_i} = \sum_{i=1}^{K} P_i ln\frac{P_i}{H_i} \qquad (13)$$

where P and H depict two one-dimensional distributions of unconditional (i.e., category) parameters, $P_i = P(x|x = i)$ and K depict a set of all probable outputs. Here P and H depict current and historical tweet information distribution. Their divergence will be small when they are identical; since $ln\left(\frac{P_i}{H_i}\right) \approx 0$ [20]. The K-L divergence can be measured by segmenting the input into categorical sets for numerical parameters. This work estimates it for every dimension of multidimensional parameters and uses cosine distance metrics for aggregating the cumulative difference using the following equation

$$d_{cos} = (P, H) = 1 - \frac{< P, H >}{\|P\|\|H\|} \qquad (14)$$

where $\|\cdot\|$ Depicts the L2 norm of a vector, and $< P, H >$ depicts the inner product of the vectors. Using Eq. (13) and Eq. (14), the model can estimate the variance between current and historical spam. If the outcome is significantly higher, then there exists a drift in session instance W. The drift time W is optimized by considering the following hypothesis

$$H = |\beta_T - \beta_U| \leq \alpha \qquad (15)$$

The parameter α is considered to be dynamic concerning the parameter β_T And optimize it according to the drift point in the preceding test window. If the outcome continues to be positive, in such a case, there exists a drift point in the preceding test window. Further, to identify the exact drift point, the model optimizes the parameter ω through empirical study. In this work, the T is split into T_1 and T_2; if there is a higher variance between T_1 and T_2 is seen, then T and U must be varying with soft-constraint. This work further establishes the drift point between T_1 and T_2; If the outcome continues to be positive, in such case, the parameter n is the drift point, i.e., $n = \omega o$ with $\omega = 0.5$. The algorithm to establish drift point W is obtained using Algorithm 1.

3.1. Training the classification algorithm

Here the model employs a cross-validation (CV) mechanism selecting useful feature sets to optimize the predictive model. Here the model selects the predictive model that reduces validation error. Most of the standard models have employed K-fold CV scheme for optimizing output. In K-fold CV, the dataset is divided randomly into K subsets of identical size; then $K-1$ subsets are used for building a predictive model, and leftover subsets are used for predicting errors in the model. Finally, the K combination of predicted error is average for obtaining CV errors. Later, a grid of l suitable values is generated to establish ideal optimizing parameters to minimize CV errors. Finally, the model with minimum CV error is selected;

Here the proposed model, a hybrid CV scheme, is proposed by combining iterative cross-validation (ICV) scheme and a feature-aware cross-validation scheme to build a predictive model that minimizes prediction error considering feature importance. The MWO-XGB model employs a CV with two-layer. In layer 1, feature subsets are chosen as the main features. In layer 2, the main subset feature selected from layer 1 is used for building the final predictive model.

First, model an Iterative CV scheme by constructing multiple sets of K folds rather than constructing single K-fold sets; the single fold CV error is obtained using the following equation

$$CV(\sigma) = \frac{1}{M}\sum_{k=1}^{K}\sum_{j\in G_{-k}} P\left(b_j, \hat{g}_\sigma^{-k(j)}(y_j,\sigma)\right) \tag{16}$$

The modified iterative CV error is obtained using the following equation

$$CV(\sigma) = \frac{1}{SM}\sum_{s=1}^{S}\sum_{k=1}^{K}\sum_{j\in G_{-k}} P\left(b_j, \hat{g}_\sigma^{-k(j)}(y_j,\sigma)\right) \tag{17}$$

Then, the optimization parameter for selecting the optimal value $\hat{\sigma}$ is obtained using the following equation

$$\hat{\sigma} = \underset{\sigma\in\{\sigma_1,\ldots,\sigma_l\}}{\arg\min}\ CV_s(\sigma) \tag{18}$$

In the above equations, $P(\cdot)$ represent loss function, $\hat{g}_\sigma^{-k(j)}(\cdot)$ Represent a function for estimating coefficients, and M describes training data size. The proposed predictive model through a modified XGBoost algorithm addressed class imbalance with concept drift awareness aid in achieving higher accuracies compared with the standard predictive model through machine learning models, which is proved through the experiment in the next section.

4. Results and Discussions

Here the performance of the MWO-XGB and existing systems such as KNN, RF, XGB, and DIA-XGB is evaluated. The experiment compares the different methods' imbalanced performance and drift detection. The performance is evaluated in terms of accuracy, recall, and F-measure.

4.1. Specificity and Sensitivity Evaluation

In this section, the specificity and sensitivity using various prediction models have been evaluated. The results are shown in Figure 2. In Figure 2, the results have been compared with the existing machine learning models like Random Forest (RF), K-Nearest Neighbor (KNN), Support Vector Machine, Ensemble learning, and XGBoost. From Figure 2, it can be seen that the MWO-XGBoost model has more specificity when compared with the other existing systems. The Sensitivity/Recall of the RF and MWO-XGB have attained the same result. It shows that the MWO-XGB has good Specificity and Sensitivity/Recall.

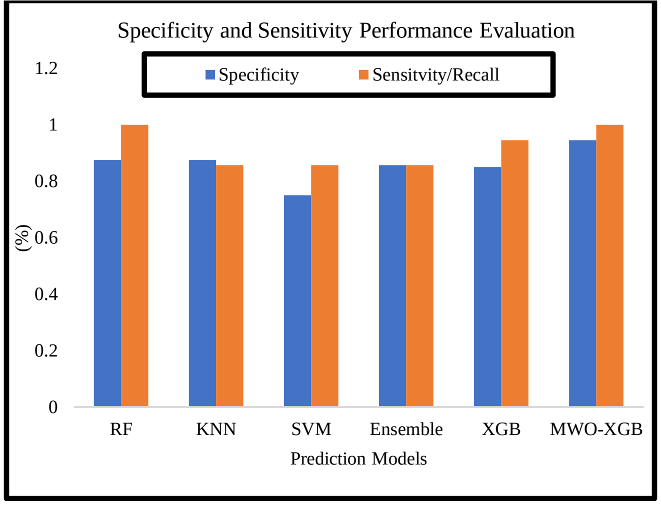

Fig. 3 Specificity and sensitivity evaluation.

4.2. ROC Performance

In this section, the ROC performance of the proposed model has been compared with the existing Ensemble and XGB models. The results have been shown in Figure 3 graphically. In Figure 3, we have compared the results using the following parameters: Specificity, Sensitivity/Recall.

Accuracy, Precision, and F-measure of the Ensemble model, XGBoost model, and our proposed MWO-XGB model. From the figure, it can be seen that the Ensemble model has less Specificity, Sensitivity,/Recall. Accuracy, Precision, and F-measure. The XGBoost model has attained better results when compared with the Ensemble model in terms of Specificity, Sensitivity,/Recall. Accuracy, Precision, and F-measure. The proposed MWO-XGB model has shown better performance in terms of Specificity, Sensitivity,/Recall. Accuracy, Precision, and F-measure compared with the Ensemble and XGB model.

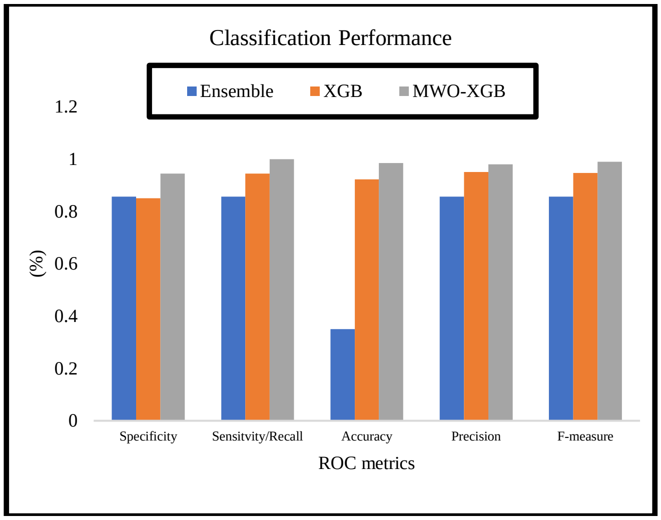

Fig. 4 Malicious URL Recall Performance

4.3. Drift-time Study

In this section, the classification performance with different drift-time has been evaluated. The parameters considered for evaluation are Specificity, Sensitivity,/Recall. Accuracy, Precision, and F-measure. The following parameters have been considered to compare the MWO-XGB with the XGBoost model. Furthermore, in Figure 4, the classification performance with drift-time=2 has been considered. The MWO-XGB model has outperformed the existing XGBoost model in terms of Specificity, Sensitivity,/Recall. Accuracy, Precision, and F-measure.

Similarly, the classification performance with drift-time=4 has been considered in Figure 5. In this also our MWO-XGB model has outperformed the existing XGBoost model. Furthermore, the classification performance with drift-time=6 and the classification performance with drift-time=8 have been carried out in Figure 6 and Figure 7, respectively. In all the classification performances having different drift-time, the MWO-XGBoost has outperformed the existing XGBoost model.

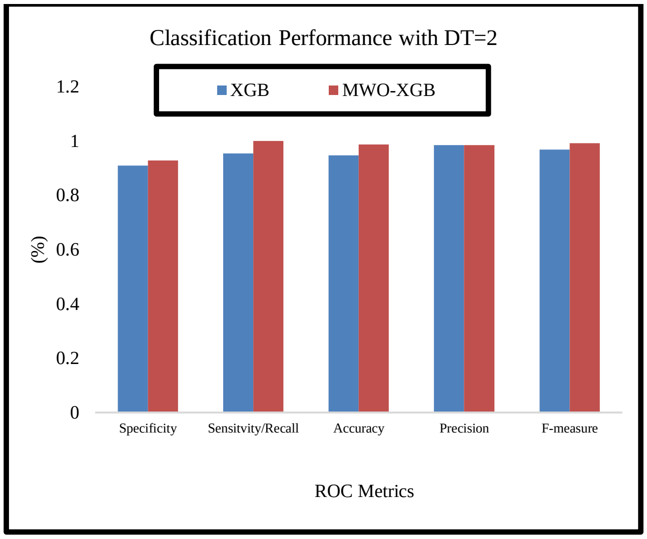

Fig. 5 Classification outcome with drift time set to 2 days.

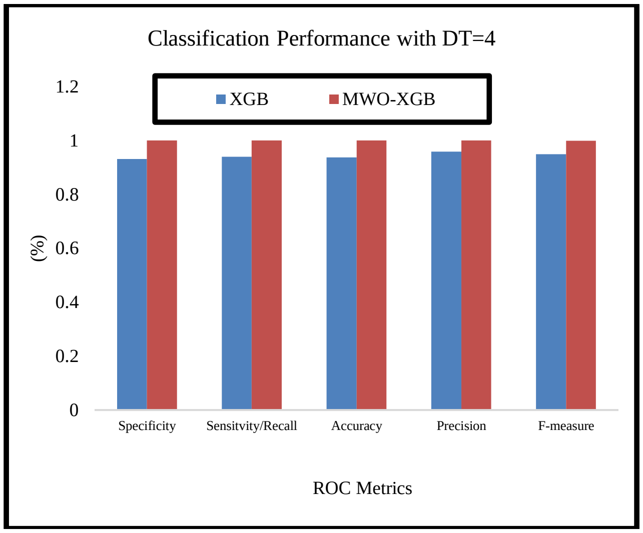

Fig. 6 Classification outcome with drift time set to 4 days.

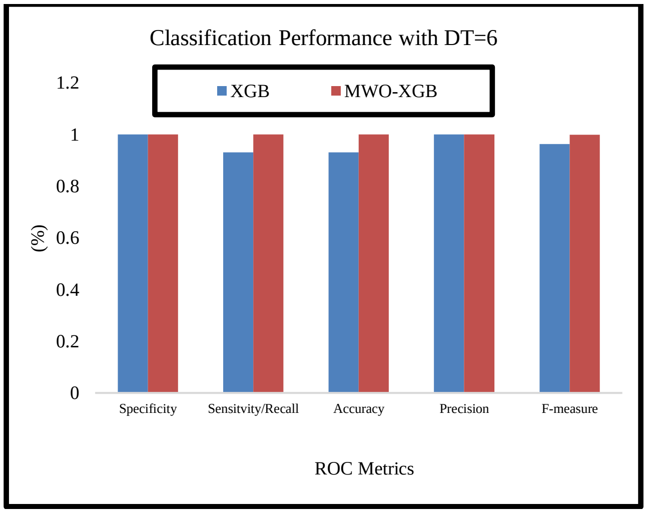

Fig. 7 Classification outcome with drift time set to 6 days.

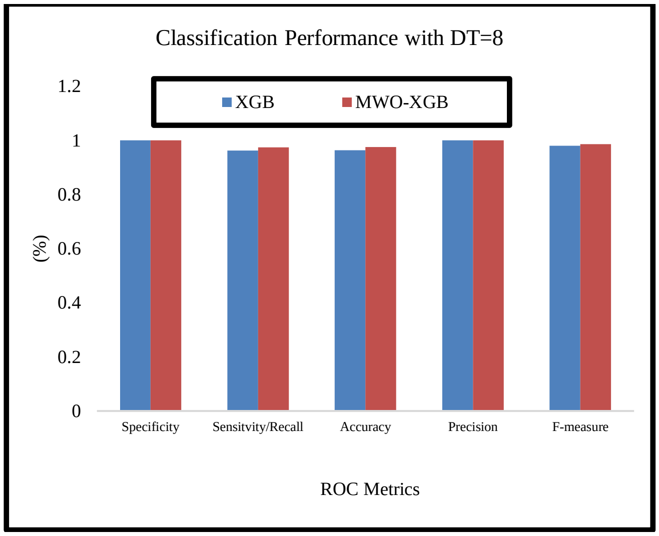

Fig. 8 Classification outcome with drift time set to 8 days.

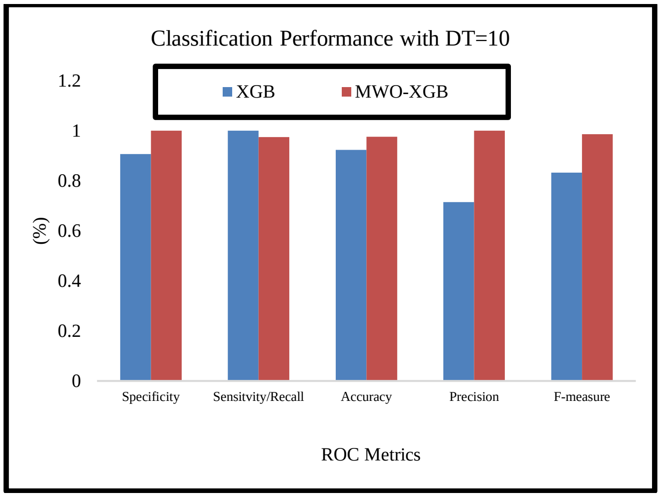

Fig. 9 Classification outcome with drift time set to 10 days.

5. Conclusion

In this paper, the researchers have first surveyed various machine learning algorithms currently being used to detect the various attacks. Further, researchers have also surveyed the concept drift and data imbalance problems. Researchers have designed a model which provides better security in an online environment using the Modified Weight Optimized XGBoost model. Experimental results have been carried out in terms of Specificity, Sensitivity,/Recall. Accuracy, Precision, and F-measure. The results have been evaluated using the social media dataset. The results show better results when compared with the existing systems. The future would study performance evaluation considering diverse attack datasets. Alongside, considers establishing which feature impacts classification accuracies.

Funding Statement

Any organization did not fund this research work.

References

[1] Anthi, Eirini & Williams, Lowri & Rhode, Matilda & Burnap, Pete & Wedgbury, Adam., Adversarial Attacks on Machine Learning Cybersecurity Defences in Industrial Control Systems. Journal of Information Security and Applications, 58 (2021) 102717. 10.1016/J.Jisa.2020.102717.

[2] M. Bagaa, T. Taleb, J. B. Bernabe and A. Skarmeta, A Machine Learning Security Framework for Iot Systems, in IEEE Access, 8 (2020) 114066-114077. Doi: 10.1109/ACCESS.2020.2996214.

[3] Sarker, Iqbal., Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science. 2 (2021). 10.1007/S42979-021-00592-X.

[4] Ligthart, Alexander & Catal, Cagatay & Tekinerdogan, Bedir., Analyzing the Effectiveness of Semi-Supervised Learning Approaches for Opinion Spam Classification. Applied Soft Computing, (2021). 101. 107023. 10.1016/J.Asoc.2020.107023.

[5] Zhou, Yuyang & Cheng, Guang & Jiang, Shanqing & Dai, Mian., Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier. Computer Networks, (2020). 174. 10.1016/J.Comnet.2020.107247.

[6] Museba, Tino & Nelwamondo, Fulufhelo & Ouahada, Khmaies., An Adaptive Heterogeneous Online Learning Ensemble Classifier for Nonstationary Environments. Computational Intelligence and Neuroscience, (2021). 2021. 1-11. 10.1155/2021/6669706.

[7] X. Liu Et Al., NADS-RA: Network Anomaly Detection Scheme Based on Feature Representation and Data Augmentation, in IEEE Access, 8 (2020) 214781-214800. Doi: 10.1109/ACCESS.2020.3040510.

[8] Sahoo, Somya Ranjan & Gupta, B., Classification of Spammer and Nonspammer Content in Online Social Network Using Genetic Algorithm-Based Feature Selection. Enterprise Information Systems, 14 (2020) 1-27. 10.1080/17517575.2020.1712742.

[9] Barushka, Aliaksandr & Hájek, Petr., Spam Detection on Social Networks Using Cost-Sensitive Feature Selection and Ensemble-Based Regularized Deep Neural Networks. Neural Computing and Applications., 32 (2020). 10.1007/S00521-019-04331-5.

[10] F. Masood Et Al., Spammer Detection and Fake User Identification on Social Networks, in IEEE Access, 7 (2019) 68140-68152. Doi: 10.1109/ACCESS.2019.2918196.

[11] Washha, Mahdi & Qaroush, Aziz & Mezghani, Manel & Sedes, Florence., Unsupervised Collective-Based Framework for Dynamic Retraining of Supervised Real-Time Spam Tweets Detection Model. Expert Systems with Applications, 135 (2019). 10.1016/J.Eswa.2019.05.052.

[12] Abkenar, Sepideh & Haghi Kashani, Mostafa & Akbari, Mohammad & Mahdipour, Ebrahim., Twitter Spam Detection: A Systematic Review, (2020).

[13] X. Wang, Q. Kang, J. an and M. Zhou, Drifted Twitter Spam Classification Using Multiscale Detection Test on K-L Divergence, in IEEE Access, 7 (2019) 108384-108394. doi: 10.1109/ACCESS.2019.2932018.

[14] X. Wang, Q. Kang, M. Zhou, L. Pan and A. Abusorrah, Multiscale Drift Detection Test to Enable Fast Learning in Nonstationary Environments, in IEEE Transactions on Cybernetics, 51(7) (2021) 3483-3495. doi: 10.1109/TCYB.2020.2989213.

[15] Yang, Li & Shami, Abdallah., A Lightweight Concept Drift Detection and Adaptation Framework for IoT Data Streams.

[16] Wahab, Omar., Sustaining the Effectiveness of IoT-Driven Intrusion Detection over Time: Defeating Concept and Data Drifts, (2021). 10.36227/techrxiv.13669199.

[17] Mehmood, Hassan & Kostakos, Panos & Cortés, Marta & Anagnostopoulos, Theodoros & Pirttikangas, Susanna & Gilman, Ekaterina., Concept Drift Adaptation Techniques in Distributed Environment for Real-World Data Streams. Smart Cities, 4 (2021) 349-371. 10.3390/smartcities4010021.

[18] C. -C. Lin, D. -J. Deng, C. -H. Kuo and L. Chen, Concept Drift Detection and Adaption in Big Imbalance Industrial IoT Data Using an Ensemble Learning Method of Offline Classifiers, in IEEE Access, 7 (2019) 56198-56207. doi: 10.1109/ACCESS.2019.2912631.

[19] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou and G. Min, Statistical Features-Based Real-Time Detection of Drifted Twitter Spam, in IEEE Transactions on Information Forensics and Security, 12(4) (2017) 914-925. doi: 10.1109/TIFS.2016.2621888.

[20] B. H. Schwengber, A. Vergütz, N. G. Prates and M. Nogueira, A Method Aware of Concept Drift for Online Botnet Detection, GLOBECOM 2020 - 2020 IEEE Global Communications Conference, (2020) 1-6. doi: 10.1109/GLOBECOM42002.2020.9347990.

[21] Museba, Tino & Nelwamondo, Fulufhelo & Ouahada, Khmaies & S.A, Akinola., Recurrent Adaptive Classifier Ensemble for Handling Recurring Concept Drifts. Applied Computational Intelligence and Soft Computing, (2021) 1-13. 10.1155/2021/5533777.

[22] Yusheng Dai, Hui Li, Yekui Qian, Yunling Guo, Min Zheng, Anticoncept Drift Method for Malware Detector Based on Generative Adversarial Network, Security and Communication Networks, Article ID 6644107, 2021 (2021) 12. https://doi.org/10.1155/2021/6644107

[23] Korycki, Łukasz & Krawczyk, Bartosz., Concept Drift Detection from Multi-Class Imbalanced Data Streams, (2021). 10.1109/ICDE51399.2021.00097.