Expert Systems With Applications xxx (xxxx) xxx

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A drift aware adaptive method based on minimum uncertainty for anomaly detection in social networking

Emad mahmodi, Hadi Sadoghi Yazdi *, Abbas Ghaemi Bafghi

Department of Computer Engineering, Ferdowsi University of Mashhad, P.O. Box: 9177948974, Mashhad, Iran

ARTICLE INFO

Keywords:
Concept drift
Data stream
Fusion of experts
Online learning

ABSTRACT

The social attack is an example of the anomaly that often changed their behavior, increased data volumes, and should be detected as early as possible to minimize damage. Data streaming mining is one of the solutions, which can handle the social attacks, and adapt to the change in the anomaly data stream. In this paper, we propose Online Fusion of Experts based on a minimum uncertainty to predict the concept drift in a data stream of social network-attack. Online learning algorithms such as Linear-order algorithms and Gaussian-order algorithms employ as an expert to identify the change in the anomaly data stream. First, online learning algorithms determine the error value for each data sample when data stream enter individually. Second, OFE utilizes a maximum-posterior estimation of the error rate of online learning algorithms to generate a new input data stream. Third, the Uncertainty Error Correlation Matrix (UECM) of input data applies to real-time behavior change detection of a data stream. Performance of OFE is evaluated by related data streaming algorithms using a benchmark, and real dataset from UCI repository (NSL-KDD, ISCX, and etc.), and malicious web pages, respectively.

1. Introduction

The optimal model of the data mining approach has been still a challenging problem for computational intelligence researchers. The traditional methods to process continuous data stream have several limitations as follows: (i) data has been storing during the observation of data stream (ii) process has been doing on data storage based on appropriate parameters (iii) the entire dataset has been requiring by the static approach. There are several computational challenges in the static approach such as the size of memory usage, limited time processing capacity, extracted features with high dimensionality, and changed pattern of continuous data. The data mining method has been used to real-world problems such as processed of satellite images, GPS systems, data security events, and information retrieval in the text, etc, (Escovedo et al., 2018). Fig. 1 shows the time-evolution of a social network attacks in each year. Adaptive to the new data behavior is the most important feature of data stream approaches. So, those approaches have been tracking dynamically where the time variation of the data is important, (Babcock et al., 2002) .

In the process of data flow, a proper model has been trained on the current data stream, which is to extract the new pattern of the data stream. The online learning methods calculate the feature weights from incoming data to extract suitable values for configuration parameters in the detection process. Concept drift referred to consider a dynamic and non-stationary environment, where changes in data distribution are frequent occurrence over time. Moreover, in the concept drift environment, the data distribution does not change, but the difference in the conditional distribution is given by arrival data. In the real-world example, concept drift happens due to the users' attention to a certain topic in the social network which has been changed constantly. So, the conditional distribution of users' attention with the re-distributive impact of data is continuously shifted from one subject to another. An adaptive learning approach has been proposed to find the immediate reaction to a concept drift which is divided into two groups: single and ensemble approach, (Farid et al., 2013).

A model for the sequence of arriving data was needed in the single-base method, whereas the ensemble method combines different models to produce a meta-model simply with better predictive performance than the majority vote of the constituent parts. A decision tree, sliding windows sampling, naive Bayes, and neural networks are an example of the single-base method and AdaBoost, Random Forest, and clustering ensemble method are an example of the ensemble-based method. The single-based approach was often utilized by a simple model but complex model often used to the ensemble-based

* Correspondence to: Departmentof Computer Engineering, Ferdowsi University of Mashhad, Azadi Ave, Azadi Square, Mashhad, Razavi Khorasan Province, Zip Code: 9177948974, Iran.

E-mail addresses: emad.mahmodi@mail.um.ac.ir (E. mahmodi), h-sadoghi@um.ac.ir (H.S. Yazdi), ghaemib@um.ac.ir (A.G. Bafghi).

https://doi.org/10.1016/j.eswa.2020.113881
Received 16 July 2019; Received in revised form 4 August 2020; Accepted 11 August 2020
Available online 19 August 2020

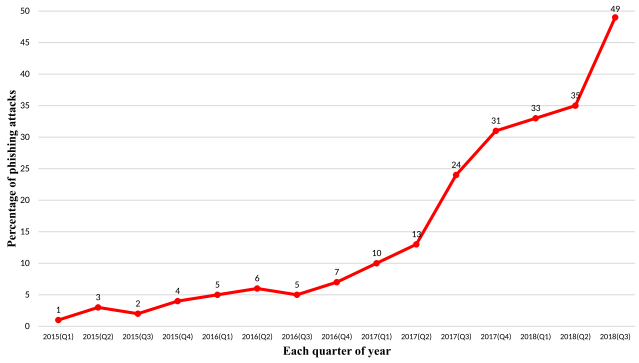

Fig. 1. Percentage of social network attacks (phishing) in each quarter of year 2015–2018.

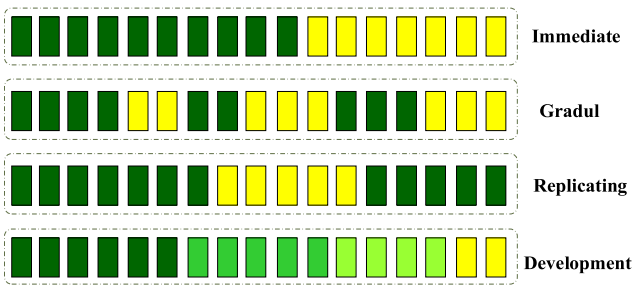

Fig. 2. The type of concept drift in an incoming data stream.

method. Also, streaming-data mining can be more tractable by efficient classifiers, (Krawczyk et al., 2017).

The importance of adaptive learning algorithms comes from the fact that it can quickly detect the concept drift. In this paper, online learning algorithms employed to both detect behavior change in a data stream and manage resource storage. We propose a drift aware adaptive method to track concept drift in order to manage behavior change in a data stream. The model takes online learning algorithms as experts and used a fusion approach to improve the performance of the experts' perspective. The principal research contributions of our system are summarized as follows:

- A new drift aware adaptive method is proposed based on the fusion of online learning algorithm to generate an optimum model of decision making without the need for additional resources. The method will be continuously combining an online learning perspective in a sliding window and can achieve better prediction accuracy.
- The existed fusion methods depend on several data mining properties such as an initial parameter of the classifier, a period of input data, several sliding windows, several classification layers, and classifiers, etc. This type of method is not effective enough to find all concept drift data stream.
- The proposed method does not re-train the online learning algorithms when the behavior change in the characteristics of the social data stream. Moreover, it does not have a computational and time overhead and suitable for the real-time environment where the social data stream has been increasing steadily.

The rest of the paper is organized as follows: Section 2 includes a review of an existing related method for concept drift detection. Section 3 a new drift aware adaptive method called OFE is proposed, in which a minimum uncertainty of online learning algorithms are employed to analyze data stream. Section 4 OFE is evaluated with artificial and real word social-attack data stream. Finally, conclusion is presented in Section 5.

2. Related work

A different type of concept drift has been happening in the real world. For example, machine learning approaches for anomaly detection in the social network have endeavored to capture the most significant anomaly in the data stream, includes attacks carried over network traffic, which it is continuously changed their strategies (Gupta et al., 2017; Liang et al., 2017; Rao & Pais, 2018; Somasundaram & Reddy, 2019). The core argument of mining algorithms in a concept drift environment can state as follows: (1) immediate drift, to recognize the quick change of the current stream. (2) Gradual drift, to define the slow change in the data distribution when a re-duplicative bag of input data is produced. (3) Replicating drift, to recognize an equal interval pattern of a data stream is repeated permanently. (4) Development drift, to detect a uniform incremental change in the pattern of the data stream. All the pattern of concept drift shows in Fig. 2.

Concept drift application related to the social network data stream includes the spam filtering, anomaly network traffic, malicious web page, where data steam with a high ratio has been arriving into the detection system, (Lane & Brodley, 1998). Research in spam detection is divided into two groups as follows: (1) a group related to messages, which are sent to the user via a social network or email. (2) Another group aims to mimic the target company's web site when the attacker creates a malicious web site similar to original ones, (Amrutkar et al., 2017). Fdez-Riverola et al. proposed a spam detection system based on instance-based reasoning (IBR) technique to handle concept drift tracking in the data stream, (Fdez-Riverola et al., 2007). Their system is applied data block to detect the distribution changing of text features in a website, which includes title, body, etc. Also, they employed the Bayesian theory to calculate in the time series when customers open a legitimate or malicious web site. The computational overhead due to the sliding windows movement and re-training phase is increased. Hsiao et al. proposed an optimal threshold and combination of different clustering algorithms called ICBC to an online classified data stream, (Hsiao & Chang, 2008). The ICBC divided a spam email data set into several groups of clusters, then employed an incremental learning technique to handle the concept drift in each individual group. Most of the online learning algorithms have been proposed to model the sequential data with timelines requirements (Ma et al., 2011). This type of algorithm required parameter optimization to obtain a simplified model of concept drift of malicious web page (Ma et al., 2011). The static sliding windows method used to many stream-processing systems, which is not suitable for the social network because of not adaptively capturing the concept drift in the data streams. Many stream-processing systems utilized the sliding windows method to capture the concept drifts over the entire a data space (Sun et al., 2017; Torquati et al., 2017). An important problem to identify concept drift in data of sliding windows is the size observation windows when different concept drift is caused by a gradual change in the data stream (Ross et al., 2012).

Domingos et al. are proposed a Very Fast Decision Tree (VFDT) to produce the slider windows alignment when quick movement applied by complex data without the use of large number data set, (Domingos & Hulten, 2000). Hulten et al. proposed the system based on the CVFDT algorithm to cope with concept drift. Their system used the decision tree to model the streaming data, whenever to change the data stream, the newest subtree is replaced to the previous subtree of the same level and update the node, (Hulten et al., 2001). Determine the observation data in the sliding windows always was a challenging problem however, they employed different types of fixed size observation windows like Hoeffding bound to examine the newest nodes that have been observed at the previous level of the tree. Hoeffding Adaptive Tree (HAT) and the Hoeffding Window Tree (HWT) are the alternative versions of Hoeffding bounds for incremental change of the number of sliding windows which proposed by Bifet and Gavalda

Table 1
Summary of concept drift algorithm.

Algorithm	Advantage	Disadvantage
Fdez-Riverola et al. (2007)	Instance-based reasoning (IBR) technique, Bayesian theory	Used block of data, Execution speed is low, Overhead is high
Hsiao and Chang (2008)	New cluster-based classification named ICBC, Group the spam data	Execution speed is low, Clustering-based approach cannot improve by feedback of labeled
Ma et al. (2011)	Incremental learning technique	Not adaptive, Not robust
Torquati et al. (2017)	Sliding windows, Memory allocate	Receiving in chunk of slid, Not adaptive
Ross et al. (2012)	Exponentially weighted moving average (EWMA) chart for monitoring concept drift	Adjust many parameters, Not robust
Domingos and Hulten (2000)	Decision Tree called VFDT, Slider windows	Receiving in chunk of slid, Not adaptive, Not robust
Hulten et al. (2001)	Decision Tree called CVFDT	Execution speed is low, Overhead is high, Not adaptive
Bifet and Gavalda (2009)	Decision Tree consist of Hoeffding Adaptive Tree (HAT) and Hoeffding Window Tree (HWT)	Not suitable for online data, Execution speed is low, Overhead is high
Mena-Torres and Aguilar-Ruiz (2014)	Adaptive learning, which proposed similarity-based method called SimC	Execution speed is low (because of execution time to define suitable labeled data by active learning is low), Overhead is high because of sampling in each step
Tennant et al. (2017)	Nearest Neighbor (NN) Clustering approach called Micro-Cluster Nearest Neighbor (MC-NN)	Clustering-based approach cannot improve by feedback of labeled, Adjust many parameters, Execution speed is low
Gama et al. (2004)	combining some online algorithm called DDM	Not robust, Execution speed is low because of cooperative learning
Baena-García et al. (2006)	combining some online algorithm called EDDM	Adjust many parameters, Not robust, Not adaptive
Proposed method	**combining Linear and Gaussian order online algorithms, relation between all online learner, Slider windows**	**adaptive, robust, Execution speed is high, Overhead is low** $O(1)$

(2009). Mena and Aguilar proposed a similarity-based method called SimC to classify the time series data, (Mena-Torres & Aguilar-Ruiz, 2014). In their system, an instance-based technique applied where implicit concept drift happens. Functional calculations of SimC is based on some requirement to choose a new sample. (1) Each class of data stream should be grouped by attribute into the individual cluster, (ii) distance between each group is calculated from the coordinates of cluster center (iii) systematically eliminated the older group by age calculation, when prediction accuracy has been developed.

Utilize of clustering approaches to handle concept drift in the streaming data have been increased. Tennant et al. proposed a new technique called Micro-Cluster Nearest Neighbor (MC-NN) to the classified data stream. The structure of their system is more complicated than the Nearest Neighbor clustering approach, which is created micro-cluster of a fixed data stream to continue the adoption of concept drift (Tennant et al., 2017). Moreover, each micro-cluster includes three terms: the label of cluster, error threshold in a cluster, and performance of micro-cluster.

Prez-solano investigated real-time sliding windows and proposed a linear regression based on the Mean Square Error method to dynamically determine the sliding windows, (Perez-Solano & Felici-Castell, 2015). Wang et al. proposed a system based on Convolution Neural Network and real-time sliding windows where time variability of historical observation of data stream is created, (Wang et al., 2012). Smrity and et al. proposed an anomaly detection scheme for the healthcare system, which is analyzed the digital signal of a data stream, (Nair & Balakrishnan, 2018). Their system based on weighted the movement of the sliding window. The experimental result shows the dynamic sliding windows system is suitable for the false alarm rate, but the system cannot reliable to detect the gradual concept drift data. Also, Deypir et al. proposed a system to identify the concept drift environment with different fluctuation of data rate in a real-time data stream, (Deypir et al., 2012). They randomly set the size of sliding windows in the initial phase, then the system adjusted the size according to the input data rate.

Gama et al. introduce the method called DDM for combining some online learning algorithms such as perceptron, a neural network, and a decision tree, then the feedback of each online learning algorithm applied to the combination procedure, (Gama et al., 2004). Baena et al. proposed a new approach (EDDM) same as the DDM, but their system has two different factors: (i) employed the online learning and offline learning algorithms (ii) combine classifier based on the distance of each classification error, (Baena-García et al., 2006). Ross et al. proposed the exponentially weighted moving average called EWMA to model the classification error rate where the concept drifts the environment (Ross et al., 2012). In their system, an online learning algorithm employed to the exploration of the data features for pattern discovery to generalized solutions of concept drift problems. The main component of EWMA for concept drift is the sequence of Bernoulli random variables. We define some of the considerable approaches in Table 1.

3. The proposed system

A significant issue in the concept drift detection methods is the functional reliability algorithms, which have a minimum delay and minimum storage cost for calculating large-scale data stream. Our goal

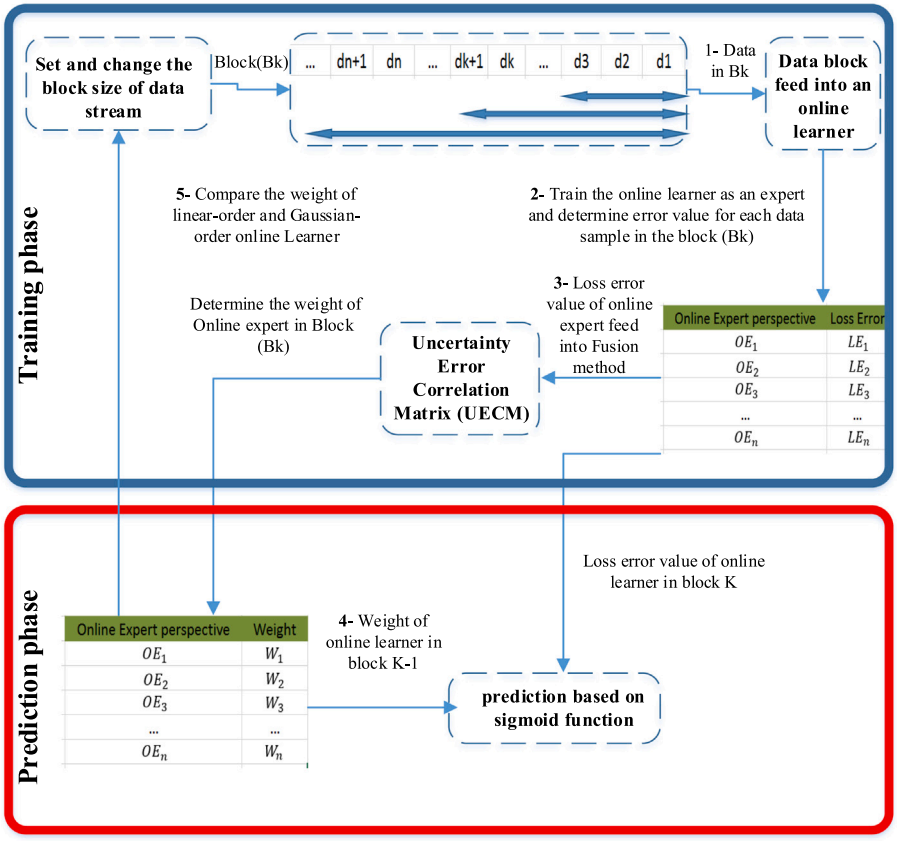

Fig. 3. Overview of the proposed drift aware adaptive method to anomaly detection in a data stream.

in this section is to describe the type of online learning algorithms as online experts and employed experts' perspective for data stream detection. We assume that all of the input data stream construct the dataset D where $D = \{(I_t, L_t)|I_t \in R, L_t \in R, t = 1, 2, \ldots, n\}$ and I_t is each data sample in time t and L_t is the label of sample data. The overall view of the proposed system is outlined in Fig. 3. In this architecture, there is six number of phases for the concept drift detection in the data stream, which is defined as follows:

- The entire data set divided into a different block, so b fix size of data blocks is created. The size of the data block in time t depends on the combining approach in the time $t-1$. In this step, our objective is to find an optimal size of the data block for the prediction model of streaming observation over time.
- Each of the data block D_k feed into the online learning algorithm individually. In this step, we employed the two types of linear-order and Gaussian-order online learning algorithms. Each of the online learning algorithms is a different mechanism to determine the concept drift adaptively.
- After training data block by an online learning algorithm, the optimal model of each online learning algorithm is achieved. So, each of the models has an error rate value in time t in data block b. In this step, the loss error value of each online learning algorithm is computed for a small portion of data.
- The result of the loss error value of the online learning algorithm in block b log into the Uncertainty Error Correlation Matrix (UECW) model, then determine the weight of each online learning algorithms.
- This step is to detect the anomaly in the data block k based on the weight of each online learning algorithm in block $k-1$ that calculated by UECW and loss error value of the online learning algorithm in the previous block $k-1$.
- Finally, the size of the next block is computed based on the loss error of each online learning algorithm in the previous block, which is calculated as follows:

$$\begin{cases} Decrease, & \text{if } (W_{(linear-order,Block_{k-1})} > W_{(Gaussian-order,Block_{k-1})}). \\ Increase, & \text{if } (W_{(linear-order,Block_{k-1})} < W_{(Gaussian-order,Block_{k-1})}). \\ Stable, & \text{if } (W_{(linear-order,Block_{k-1})} = W_{(Gaussian-order,Block_{k-1})}). \\ 0, & \text{otherwise.} \end{cases}$$

3.1. Online expert

To use the online learning algorithms two scenarios are employed. The first is $Linear-order$ online learning algorithms, which the classification algorithms effects from some linear parameter. The second is $Gaussian-order$ online learning algorithms, which all of the classification parameters influenced by Gaussian distribution in a data stream. Both of the online learning algorithms depend on a data stream, in which each input data individually enter then, employs a different mechanism for concept drift, which is not always the best performance. The Fig. 4 represent the overall view of the OFE.

Perceptron is an online learning algorithm, which is based on the feedback the *Perceptron* received from one hidden layer from streaming data (Irie & Miyake, 1988). The other kind of similar online learning algorithm is OGD, which based on minimum the gradient of input data (Zinkevich, 2003). The problems of *Perceptron* and OGD are the weakness of incremental and sudden concept drift detection, respectively. Hence, other types of online learning algorithms such as Passive-aggressive (PA) are used to improve the performance of concept drift detection (Crammer et al., 1993). Online learning algorithms consider the different types of optimization problems to produce a better model of a data stream over time.

$$W \in R, w_{t+1} = argmin\frac{1}{2}\|W - w_t\|, st : (w_t'.I_t.L_t) > 0 \quad (1)$$

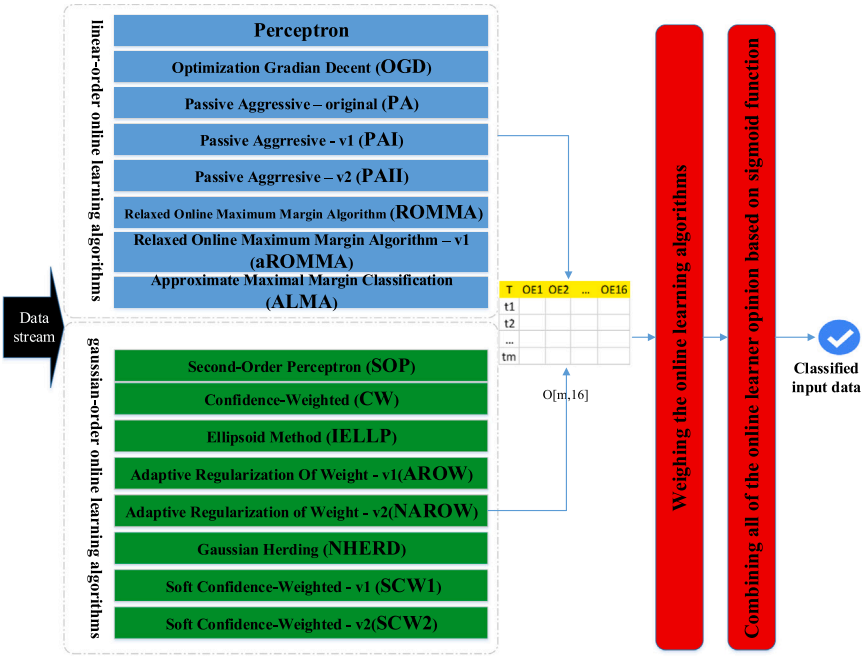

Fig. 4. The architecture of combining the online learners' perspective.

Data: D
Result: Predict label and update parameter
Initialization w_0;
while $t \in Time$ **do**
  Each data in data stream is received: $I_t \in InputData$;
  True class label was defined: $L_t \in Label$;
  while $OE(Online\ Expert) \in Online\ learning\ algorithms$ **do**
    Class label of OE_i was predicted: $\hat{L_{OE_i}} = sign(l(f(I_t.w_t)))$;
    Loss function of OE_i was calculated: $Loss_{OE_i} = l(w_t.(I_t.L_t))$;
    if $l(w_t.(I_t.L_t)) > 0$ **then**
      The learner updates the classification model:
      $w_{t+1} = w_t + \triangle(w_t.(I_t.L_t))$;
    else
      $w_{t+1} = w_t$;
    end
  end
  Fusion online expert $(\hat{L_{OE_i}}, Loss_{OE_i}, L_t)$;
end

Algorithm 1: Fusion of online expert's prediction

Loss function and input data value are important driving factors to consider the optimal model for concept drift detection. In each round, the model trend towards the best weight incrementally. By solving the optimization problem Eq. (1), PA updates the model equivalent as the following:

$$w_{t+1} = w_t + \frac{l_t}{\|I\|^2}\gamma L_t I_t \tag{2}$$

where w_{t+1} is a new weight of the input data stream; l_t is a loss function of the prediction for estimates the status of the current sample. γ is a parameter that determines the importance value of loss function. Various states of concept drift are considered by the cost function and the other parameter like loss value. Some types of $linear-order$ online learner for concept drift shows in Table 2.

The Gaussian-order is the other type of online learning algorithms, which updates their parameter based on Gaussian distribution. The mean value $\mu \in R$ and covariance matrix $\Sigma \in R$ of input data distribution is the principal factor in concept drift detection. Confidence weighting (CW) is one of the most popular online learning algorithms, which is based on Gaussian-order (Dredze et al., 2008) online learning algorithm. For a data stream, when a low variance of data is obtained, CW provides more confidence to result in the average data. To recognize the concept drift in a data stream, divergence Kullback Leibler (KL) is used as follow:

$$(\mu, \Sigma) = argmin D_K L(\mathcal{N}(\mu, \Sigma) \parallel \mathcal{N}(\mu_{t+1}, \Sigma_{t+1})), st : (w_t'.x_t.y_t) > 0 \tag{3}$$

After the solving of cost function Eq. (3), the new weighting of each input data calculated as follow:

$$w_{t+1} = w_t + \alpha_t.x_t.y_t.\Sigma_t' \tag{4}$$

$$\Sigma_{t+1} = \Sigma_t - \beta_t.x_t.\Sigma_t' \tag{5}$$

In these approaches, the significant parameters are α_t, β_t, which depends on weight and Gaussian distribution of feature's input data, respectively. These are more efficient for computing loss function. Another type of Gaussian order online experts represented to handle other types of concept drift, which shows in Table 3.

3.2. Online Fusion of Expert (OFE)

This section attempt to model the dynamic changes in the concept drift data stream. The online Fusion of Expert (OFE) is considered the type of concept drift data stream. It also can continually adaptive to the behavior change in the data stream when a different concept is activated. As shown in Tables 2, 3, each of the OE has been utilized a different mechanism to concept drift detection. Also, none of the online experts always have the highest performance for the type of concept drift all the time. OFE takes into account the effect of each online learner Observation.

The benefit of OFE is the capability of the method for process a large number of the data stream. Also, OFE able to adaptive with the distribution of the extracted features over time.

Definition 3.1. Dataset $\langle D \rangle$ is an infinitive order of the data which keep changing. In all accessibility data set D, each sample data $\langle I_t \rangle$ follow the schema $\overrightarrow{d_i} = \langle F_1, F_2, \ldots, F_m \rangle$ has d-dimensional features. We employ our weighting feature set $\langle F \rangle$ for adaptive to a concept drift

Table 2
Types of *Linear−order* online learning algorithm and the effectiveness of online method as concept drift detection ($\gamma, \alpha, \beta, c, p, k$ are constant variable).

Algorithm	Loss function	Weight update	Concept drift
Perceptron (Irie & Miyake, 1988)	$l_t = max(0, 1-L_t w_t I_t)$	$w_{t+1} = w_t + l_t L_t I_t$	immediate
Stochastic Gradient Descent (OGD - logistic loss)(Zinkevich, 2003)	$l_t = log(1 + e^{(-L_t w_t I_t)})$	$w_{t+1} = w_t + \gamma l_t L_t I_t$	replicating
passive aggressive (PA) (Crammer et al., 1993)	$l_t = max(0, 1-L_t w_t I_t)$	$w_{t+1} = w_t + \frac{l_t}{\|I\|^2} \gamma L_t I_t$	gradual
passive aggressive (PA1) (Crammer et al., 1993)	$l_t = max(0, 1-L_t w_t I_t)$	$\gamma_t = min(c, \frac{l_t}{\|I\|^2})$, $w_{t+1} = w_t + \gamma l_t L_t I_t$	gradual, replicating
passive aggressive (PA2) (Crammer et al., 1993)	$l_t = max(0, 1-L_t w_t I_t)$	$\gamma_t = \frac{l_t}{\|I\|^2 + \frac{1}{2*c}}$, $w_{t+1} = w_t + \gamma l_t L_t I_t$	gradual, replicating
Relaxed Online Maximum Margin Algorithm (ROMMA)(Li & Long, 2000)	$l_t = max(0, 1-L_t w_t I_t)$	$\alpha_t = \frac{(\|I_t\| * \|w\|)^{2-(L_t w I_t)}}{(\|I_t\| * \|w\|)^{2-(w I_t)}}$, $\beta_t = \frac{\|w\|^2 * L_t w I_t}{(\|I_t\| * \|w\|)^{2-(w I_t)}}$, $w_{t+1} = \alpha_t w_t + \beta_t I_t$	immediate, replicating
aROMMA (Li & Long, 2000)	$l_t = 1 - L_t w_t I_t$	$\alpha_t = \frac{(\|I_t\| * \|w\|)^{2-(L_t w I_t)}}{(\|I_t\| * \|w\|)^{2-(w I_t)}}$, $\beta_t = \frac{\|w\|^2 * L_t w I_t}{(\|I_t\| * \|w\|)^{2-(w I_t)}}$, $w_{t+1} = \alpha_t w_t + \beta_t I_t$	immediate, replicating
Approximate Maximal Margin Classification Algorithm (ALMA) (Gentile, 2001)	$l_t = (1-\alpha) * \frac{\frac{1}{\alpha}\sqrt{p-1}}{k+1} - L_t w_t I_t$	$\beta_t = \frac{c}{\sqrt{p-1} * \sqrt{t}}$, $w_{t+1} = w_t + \beta_t L_t I_t$, $w_{t+1} = \frac{w}{max(1, \|w_t\|)}$	replicating

data stream. Our system can apply a new feature to the system without interruption during the detection procedure. When the new feature $\langle F_{t+1} \rangle$ has been extracted then, initial value of feature vector equal to zero $\langle F_1, F_2, \ldots, F_d, 0 \rangle$. In broad terms, the online learning algorithm employs to handle this problem.

Algorithm 1 illustrates the steps of our model for a data stream. The input data sequentially feed into the online learning algorithms. Since each online learning algorithm gives their perspective on input data that is not similar to another. Then Loss function of each online learner ($Loss_{OE}$) and updating parameter can use to combining all opinions of the OE. In Algorithm 1, $\triangle(w_t.(x_t.y_t))$ and $l(w_t.(x_t.y_t))$ define the update and loss function respectively.

For the streaming concept drift, we create sliding windows to determine the weight of the OE. We find the sufficient weight for the OE by sliding windows block in time t when each of the loss function is taken into the appropriate values. The data stream splitting up to the sequential block s_k, which is equal to $S = s_1, s_2, \ldots, s_p$, then each online learner determines the weighting bound of behavior change in block distribution (see Fig. 5).

Definition 3.2 (*Uncertainty Error Correlation Matrix(UECM)*. $UECM$ describes the correlation strength between the loss function of each expert. Moreover, $UECM$ lets us use the OE prediction result for added their effect. In each block, the $UECM$ is according to the $K \times K$ symmetric matrix (shown in Table 4) where weight $\sigma(UECM_{i,j})$ is the correlation of the experts' perspective, which is calculated from error rate in the block.

The model applies to the online learner loss function when the set of data loaded into the block s_i. So, The strength dependency between the OE in each block calculated as problem formulation (6).

$$P(w) = \sum_{i=1}^{N} \Psi_{i,j} E_{i,j} \tag{6}$$

Which, the prediction value of expert E_k for data stream I_t combine to their weight $\Psi_{i,j}$. In general, this requires changed the weight for each expert, whereas the new drift has seen in the data stream. This problem modeled by the correlation between the error rates of the experts, and calculated as follow:

$$F_t = \sum_{i=1}^{L} \sum_{i=1}^{L} \Psi_i \Psi_j \sigma_{i,j} - \lambda(\sum_{i=1}^{l} \Psi_i - 1) \tag{7}$$

and when solving the optimization problem Eq. (7) we can access to the best weight as an Eq. (8)

$$\Psi_t = \Sigma_t^{-1} I (I^t \Sigma_t^{-1} I)^{-1} \tag{8}$$

In Eq. (7), $\sigma_{i,j}$ determine the input-correlation between expert i and j based on covariance error of their view. Finally, with respect to the minimum of cost function F, the value of Ψ is obtained from (8), which $\Psi = [\Psi_1, \Psi_2, \ldots, \Psi_k]$ is the prediction weight of each experts' view, also the value must be estimated by covariance matrix (Σ_t) of expert's error rate in the block.

For the calculation of Σ_t explained in Eq. (8), we are using the zero-mean value of the error hence we required the mean of the data as to the determination of their covariance. So, $\bar{I}_t$ calculated based on maximum posterior (MAP). The Σ_t calculated based on parameter $\bar{I}_t$ of Gaussian-distributed that has variance $\sigma_{\bar{I}_t}$ and mean $u_{\bar{I}_t}$ as following:

$$P(\bar{I}_t) = \frac{1}{(2\pi\sigma_t^2)^{1/2}} exp[-\frac{(\bar{I}_t - u_{\bar{I}_t})^2}{2\sigma^2}] \tag{9}$$

By employing the likelihood function and the prior pdf, we can define the posterior pdf based on Bayes theory as follows:

$$P(\bar{I}_t | I_{1:t}) = \frac{1}{p(I_{1:t})} p(I_{1:t} | \bar{I}_t) p(\bar{I}_t) \tag{10}$$

$$P(\bar{I}_t | I_{1:t}) = \frac{1}{p(I_{1:t})(2\pi\sigma_n^2)^{N/2}(2\pi\sigma_n^2)^{1/2}} exp[-\frac{1}{2\sigma_n^2} \sum_{m=0}^{N-1} (I_m - \bar{I}_t)^2$$

Table 3
Types of $Gaussian-order$ online learning algorithm.

Algorithm	Loss function	Weight update	Concept drift
Confidence-Weighted Learning (CW) (Dredze et al., 2008)	$l_t = max(0, 1-L_t w_t I_t)$	$\alpha_t = max(0, \frac{\gamma L_t w_t I_t + \sqrt{(L_t w_t I_t)^2\gamma^4 + 4 I_t^2 \Sigma \gamma^2}}{\gamma I_t^2 \Sigma})$, $\mu_t = 0.25(-\alpha_t I_t^2 \Sigma\gamma + \sqrt{(\alpha_t^2 I_t^2 \Sigma\gamma)^2 + 4 I_t^2 \Sigma)^2}$, $\beta_t = \frac{\alpha_t \gamma}{\mu_t + \alpha_t I_t^2 \Sigma\gamma}$, $w_{t+1} = w_t + \alpha_t L_t I_t \Sigma$, $\Sigma_{t+1} = \Sigma_t - \beta(I_t \Sigma_t)^2$	Gradul, Replicating, Development
Soft Confidence-Weighted Learning (SCW) (Wang & Li, 2018)	$l_t = \gamma\sqrt{x_t^2 \Sigma - L_t w_t I_t}$	$\alpha_t = max(0, \frac{\gamma L_t w_t I_t + \sqrt{(L_t w_t I_t)^2\gamma^4 + 4 I_t^2 \Sigma \gamma^2}}{\gamma I_t^2 \Sigma})$, $\alpha_{t+1} = min(\alpha_t, C)$, $\mu_t = 0.25(-\alpha_t I_t^2 \Sigma\gamma + \sqrt{(\alpha_t^2 I_t^2 \Sigma\gamma)^2 + 4 I_t^2 \Sigma)^2}$, $\beta_t = \frac{\alpha_t \gamma}{\mu_t + \alpha_t I_t^2 \Sigma\gamma}$, $w_{t+1} = w_t + \alpha_{t+1} L_t I_t \Sigma$, $\Sigma_{t+1} = \Sigma_t - \beta(I_t \Sigma_t)^2$	Gradul, Replicating, Development
Soft Confidence-Weighted Learning2 (SCW2) (Wang et al., 2018)	$l_t = \gamma\sqrt{x_t^2 \Sigma - L_t w_t I_t}$	$\lambda = \sqrt{(L_t w_t I_t)^2 \gamma_t^2 I_t^2 \Sigma + \frac{I_t^2 \Sigma + 1}{C}}$, $\alpha = max(0, \frac{2L_t w_t I_t + L_t w_t I_t^2 \Sigma + \lambda}{(\frac{I_t^2 \Sigma + 1}{2C})^2 + \frac{I_t^2 \Sigma + 1}{2C} * I_t^2 \Sigma\gamma})$, $\alpha_{t+1} = min(\alpha_t, C)$, $\mu_t = 0.25(-\alpha_t I_t^2 \Sigma\gamma + \sqrt{(\alpha_t^2 I_t^2 \Sigma\gamma)^2 + 4 I_t^2 \Sigma)^2}$, $\beta_t = \frac{\alpha_t \gamma}{\mu_t + \alpha_t I_t^2 \Sigma\gamma}$, $w_{t+1} = w_t + \alpha_{t+1} L_t I_t \Sigma$, $\Sigma_{t+1} = \Sigma_t - \beta(I_t \Sigma_t)^2$	Gradul, Replicating, Development
Adaptive Regularization of Weight Vectors (AROW) (Crammer et al., 2009)	$l_t = max(0, 1-L_t w_t I_t)$	$\alpha_t = (1 - L_t(w_t I_t)) \frac{1}{I_t^2 \Sigma_t + \gamma}$, $w_{t+1} = w_t + \alpha_t L_t I_t \Sigma_t$, $\Sigma_{t+1} = \Sigma_t - \frac{1}{I_t^2 \Sigma_t + \gamma} I_t \Sigma_t$	Gradul, Replicating, Development
New Adaptive Algorithms for Online Classification (NAROW) (Crammer et al., 2009)	$l_t = 1 - L_t w_t I_t$	$\beta_t = \frac{1}{I_t^2 \Sigma + \frac{I_t^2 \Sigma}{\gamma I_t^2 \Sigma - 1}}$, $\alpha_t = max(0, 1 - L_t w_t I_t)\beta_t$, $w_{t+1} = w_t + \alpha_t L_t I_t \Sigma$, $\Sigma_{t+1} = \Sigma_t - \beta_t I_t \Sigma$	Gradul, Replicating, Development
Second Order Perceptron Algorithm (SOP) (Cesa-Bianchi et al., 2005)	$l_t = max(0, 1-L_t w_t I_t)$	$\beta_t = \frac{1}{I_t \Sigma + 1}$, $\alpha_t = max(0, 1 - L_t w_t I_t)\beta_t$, $w_{t+1} = w_t + L_t I_t$, $\Sigma_{t+1} = \Sigma_t - \beta_t (I_t \Sigma)^2$	Gradul, Replicating, Development
Learning via Gaussian Herding (NHERD) (Crammer & Lee, 2010)	$l_t = 1 - L_t w_t I_t$	$\beta_t = \frac{1}{I_t^2 \Sigma + \gamma}$, $\alpha = max(0, 1 - L_t w_t I_t)\beta$, $w_{t+1} = w_t + \alpha L_t I_t \sigma$, $\Sigma_{t+1} = \Sigma_t - \beta_t^2 (I_t^2 \sigma + 2\gamma)(I_t \Sigma)^2$	Gradul, Replicating, Development
Online Learning by Ellipsoid Method (IELLIP) (Yang et al., 2009)	$l_t = max(0, 1-L_t w_t I_t)$	$\alpha_t = \frac{1 - L_t w_t I_t}{\sqrt{\Sigma I_t^2}}$, $\beta_t = \frac{L_t I_t \Sigma}{\sqrt{\Sigma I_t^2}}$, $w_t + 1 = w_t + \alpha_t \beta_t$, $\Sigma_{t+1} = \frac{\Sigma_t - \gamma \beta_t^2}{1 - \gamma}$	Gradul, Replicating, immediate

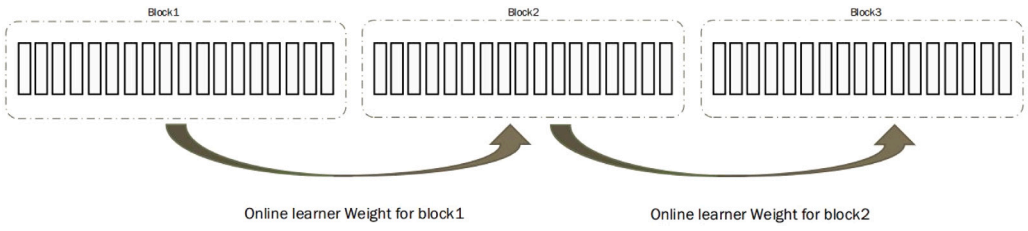

Fig. 5. OFE procedure on the data sequence.

$$- \frac{1}{2\sigma^2_{\bar{I}_t}(\bar{I}_t - u_{\bar{I}_t})^2}] \tag{11}$$

where $I_{1:t}$ is observation of input data value until time t. If we assume $\bar{I}_t$ value for the log-posterior function $(P(\bar{I}_t|I_{1:t}))$ equal to zero, so maximum posterior solution is getting as:

$$\bar{I}_{tMAP} = \frac{\sigma^2_{\bar{I}_t}}{\sigma^2_{\bar{I}_t} + \frac{\sigma^2_n}{N}} \frac{1}{N} \sum_{m=0}^{N-1} I_m + \frac{\frac{\sigma^2_n}{N}}{\sigma^2_{\bar{I}_t} + \frac{\sigma^2_n}{N}} u_{\bar{I}_t} \tag{12}$$

As shown in Eq. (12), MAP estimate is dependent on number of error and input data. By substitute of Eq. (12), we estimate the expectation of the MAP when $E[\frac{1}{N}\sum_{m=0}^{N-1} I_m] = \bar{I}_t$ as follow:

$$E[\bar{I}_{tMAP}] = \frac{\sigma^2_{\bar{I}_t}}{\sigma^2_{\bar{I}_t} + \frac{\sigma^2_n}{N}} \bar{I}_t + \frac{\frac{\sigma^2_n}{N}}{\sigma^2_{\bar{I}_t} + \frac{\sigma^2_n}{N} u_{\bar{I}_t}} \tag{13}$$

And its variance defines as follows:

$$Var[\bar{I}_{tMAP}] = \frac{\sigma^2_{\bar{I}_t}}{\sigma^2_{\bar{I}_t} + \frac{\sigma^2_n}{N}} Var[\frac{1}{N}\sum_{m=0}^{N-1} I_m] + \frac{\frac{\sigma^2_n}{N}}{\sigma^2_{\bar{I}_t} + \frac{\sigma^2_n}{N} u_{\bar{I}_t}} \tag{14}$$

And by substitution of $Var[\bar{I}_t] = E(\bar{I}_t - \bar{I})^2 = E\{[\frac{1}{N}\sum_{m=0}^{N-1} I_m]^2\} = \frac{\sigma^2_n}{N}$ into the Eq. (14), $Var[\bar{I}_t]$ taking as follows:

$$Var[\bar{I}_{tMAP}] = \frac{Var(\bar{I}_t)}{1 + \frac{Var(\bar{I}_t)}{\sigma^2_{\bar{I}_t}}} \tag{15}$$

So we can estimate zero-mean input data value over time as follow:

$$I_{t,z} = I_t - \bar{I}_{tMAP} \tag{16}$$

The Σ_t value during the period of time obtained from the following equations:

$$\Sigma_t = \sum_{i=1}^{t} \lambda^{t-i} I_{t,z} I^T_{t,z}, p_t = \sum_{i=1}^{t} \lambda^{t-i} I_{t,z} d_t \tag{17}$$

where P_t is the cross-correlation between each data, which can be an anomaly or normal social-network traffic and the desired value in time t, also λ $(0 > \lambda > 1)$ is an forget factor of influence data block. We can also write Σ_t and p_t as

$$\Sigma_t = \lambda\Sigma_{t-1} + I_{t,z} I^T_{t,z}, p_t = \lambda p_{t-1} + I_{t,z} d_t \tag{18}$$

To calculate the Σ^{-1}_t, we employ the $MATRIX\ INVERSION\ LEMMA$ that define by Haykin (1986). Based on this lemma, both of the Σ_t and p_t has a positive definition which can be determined as follow:

$$A = \Sigma_t, B^{-1} = \lambda\Sigma_{t-1}, C = I_{t,z}, D = 1 \tag{19}$$

So with the placement of variables, we can access to:

$$\Sigma^{-1}_t = \lambda\Sigma^{-1}_{t-1} - \frac{\lambda^{-2}\Sigma^{-1}_{t-1} I_{t,z} I^T_{t,z} (\Sigma^{-1}_{t-1})^T}{1 + \lambda^{-1} I^T_{t,z} \Sigma^{-1}_{t-1} I_{t,z}} \tag{20}$$

If we consider that

$$\kappa_t = \frac{\lambda^{-1}\Sigma^{-1}_{t-1} I_{t,z}}{1 + \lambda^{-1} I^T_{t,z} \Sigma^{-1}_{t-1} I_{t,z}} \tag{21}$$

So with respect to the Eq. (21), we can define the Eq. (20) as

$$\Sigma^{-1}_t = \lambda^{-1}\Sigma^{-1}_{t-1} - \lambda^{-1}\kappa_t I^T_{t,z}\Sigma^{-1}_{t-1} \tag{22}$$

and taking

$$\begin{aligned} \kappa_t &= \lambda^{-1}\Sigma^{-1}_{t-1} I_{t,z} - \lambda^{-1}\kappa_t I^T_{t,z}\Sigma^{-1}_{t-1} I_{t,z} \\ &= (\lambda^{-1}\Sigma^{-1}_{t-1} - \lambda^{-1}\kappa_t I^T_{t,z}\Sigma^{-1}_{t-1}) I_{t,z} = \Sigma^{-1}_t I_{t,z} \end{aligned} \tag{23}$$

Table 4
Uncertainty Error Correlation Matrix($UECM$) to determine the weight of experts' perspective.

	L1	..	G1	..
L1				
:				
G1				
:				

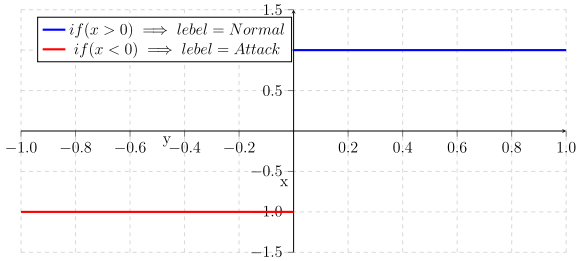

Fig. 6. The sigmoid function used to classify between the Normal and Attack data.

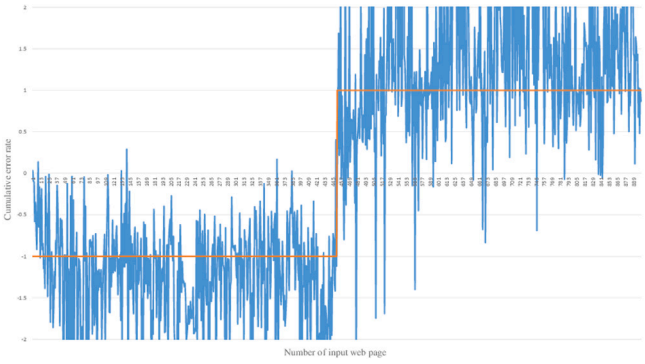

Fig. 7. The confidence value of a malicious and legitimate web page, which is classified by OFE.

Given the simplification and placement of variable, we obtain the final equation as follow:

$$\Sigma^{-1}_t = \lambda\Sigma^{-1}_{t-1} - \frac{\lambda^{-2}\Sigma^{-1}_{t-1} I_{t,z} I^T_{t,z} (\Sigma^{-1}_{t-1})^T}{1 + \lambda^{-1} I^T_{t,z} \Sigma^{-1}_{t-1} I_{t,z}} \tag{24}$$

Finally, we can define weight Ψ_t as follows:

$$\Sigma_t \Psi_t = p_t, \Psi_t = \Sigma^{-1}_t p_t, \tag{25}$$

The Fig. 7 shows that the first half of the data stream and the rest of them consider as an anomaly and legitimate traffic, respectively. The sigmoid function utilized to classify anomaly detection in the data stream, as shown in Fig. 6. In Fig. 7, the classification is based on the result of the proposed model. So, for the legitimate web pages, the prediction is upper than zero, and for malicious web pages is less than zero.

The Fig. 8 shows an example of the proposed method, each sample of the data block 1, contain n sample data, feed into the OE. The result of OE is loss error value and weight of data stream, which is used to detect the drift in the individual observation data and achieved the optimal model of W over time. The loss error value only employs as an input to our combining model. So, after executing OE by all the

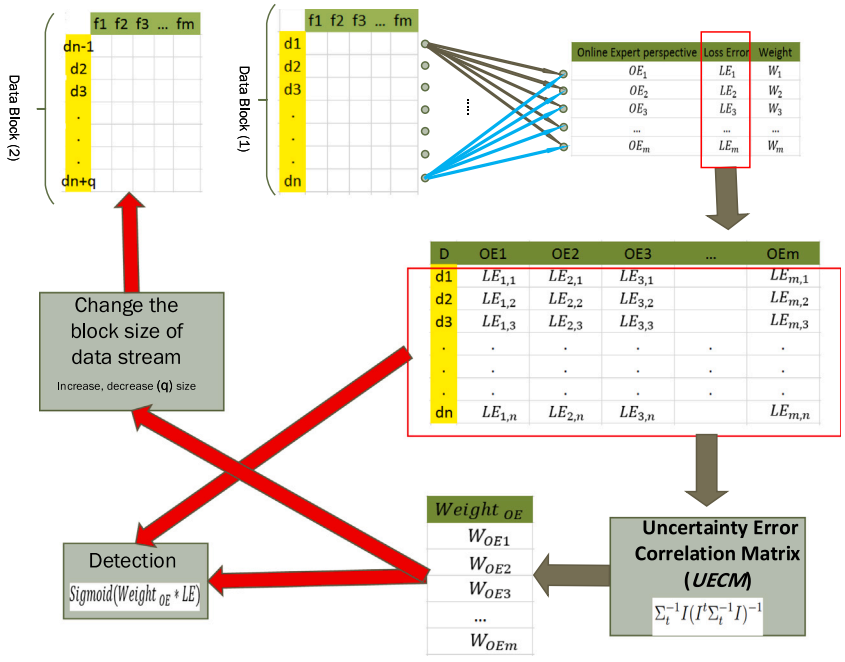

Fig. 8. An example of our architecture step bay step.

Table 5
Data statistics of each experimental dataset.

Dataset	Attacks	Normal
NSL KDD	back, buffer overflow, ftp write, guess passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster, normal, unknown	Network normal traffic
ISCX	Fuzzers, Reconnaissance, Shellcode, Analysis, Backdoors, DoS, Exploits, Worms	Network normal traffic
Malicious web page	Phish and spam web page	Alexa web page

data in the block 1, the matrix data and OE created, which is consist of n row (number of data) and m column (number of OE). This matrix applied by the UECW method (Eq. (8)) to take the weight of each OE in block 1, So we have the vector of weighted of each OE from the diagonal values of the matrix $UECW$, which the dimension is equal to m. Finally, the multiplication of $Weight_{OE}$ in block 1 with one of loss error value in block 2, produce a scalar value, which is used to for the prediction based on sigmoid function. Also, we can adjust the size of the block by $Weight_{OE}$ when the weight is changed.

4. Evaluation configuration

We evaluated the performance of our system on benchmark and real dataset, then compared it to the other simple and combining system for concept drift detection in the social-network data stream. To circumvent the streaming concept drift problems, we present two types of online learning algorithms in which the evaluation of each considered. Some of the Linear-order and Gaussian-order online learning algorithms are used the recursive form but in our work, an online streaming type is employed, which an input data continually increased. Linear-order online learner is chosen to deal with immediate and development concept drift. Sometimes time, the other type (Gaussian-order) is applied to detect development, gradual, and replicating concept drift. On the other hand, both of the online learners suitable for the computational process and reduce time consumption. The proposed system based on combining the online learner perspective about the data observation block, which is the influence on the next block.

The proposed system needs to initialize some of the parameters in online learning algorithms. So, the value of the parameters for each case of the data stream is different. So, the fundamental value of parameters is set randomly in the initial step, but it can converge to a suitable value where all of the online learning algorithms achieve a reasonable perspective. In linear-order and Gaussian-order online learning algorithm, γ and C are depict the maximum effect of error and distribution of input data, respectively. So, in the setting of values, $\gamma = 0.7, C = 2$ increased the performance of the online learning algorithm.

In our system, each online learning perspective can employ individually, so the Parallelization of each expert is sufficient for maximizing time speed. Size of the block, another influential parameter to the calculation of combining result, hence in the initil step, we used the size of block equal from 100 to 600 and achieved that the block size 200 is faster and efficient to use for concept drift. Though, if we increased the size of the block more than a threshold, it is the unfavorable impact on the detection.

Data set In this section, we evaluate our system based on the different social network data streams and investigate the performance of the related system. So, our investigation dataset divided into two overall types as follows: artificial (benchmark) and real current data set. For the detection process, we have two classes in each dataset, normal, and anomaly class label. The anomaly data incrementally occurs in the data stream, where the pattern of the data stream has been changed. (Data features shows in Table 5)

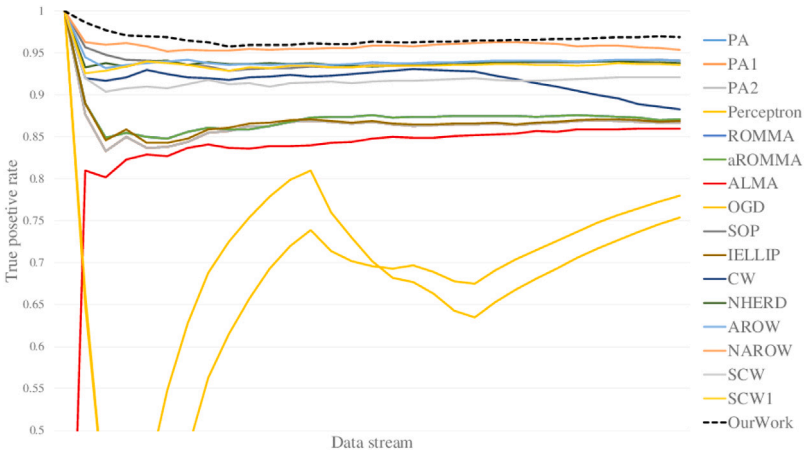

Fig. 9. True positive rate of network traffic (NSL-KDD data stream).

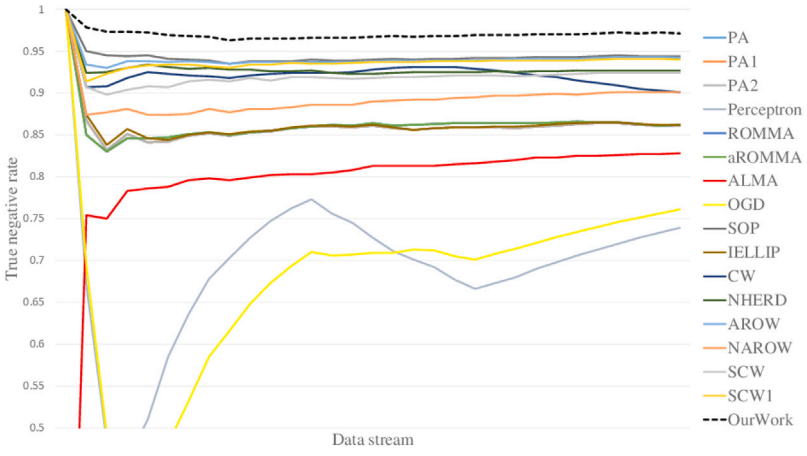

Fig. 10. True negative rate of network traffic (NSL-KDD data stream).

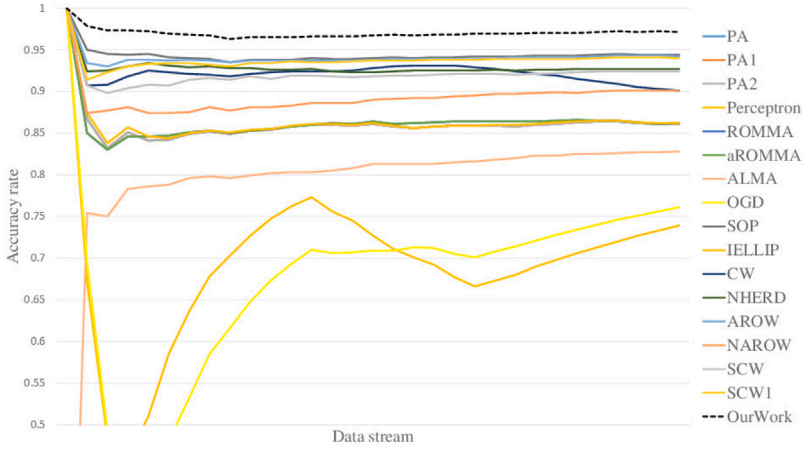

Fig. 11. Accuracy rate of network traffic (NSL-KDD data stream).

NSL KDD. NSL KDD is one of the public datasets, which applied to the intrusion detection system in the network traffic. NSL KDD is a new version of KDD99, which is created in 1999, giving a total of 125000 data and 41 feature vector. Also, there is 22 kind of attacks that depends on each sample of the data stream feature vector, (Bagheri et al.).

ISCX. Types of DoS attacks are one of the significant problems existing in the current year, which is considered in the ISCX dataset. Many new online classification approaches required of ISCX for detecting application-layer DoS attacks. The ISCX has 8 sets of DoS attacks and 24 h of network traffic, (Mamun et al., 2016).

Real data. Malicious web pages is another type of streaming data set, which is under the change of attackers who regularly create their web page by the distinctive pattern. Therefore, we need to obtain the content of the malicious web site https://www.phishtank.com/ and normal web site https://www.alexa.com/topsites/countries/US. All features of

	TPR	TNR	Accuracy	Fscore	MCC
PA	0.866161	0.861	0.861	0.868258	0.721161
PA1	0.866161	0.861	0.861	0.868258	0.721161
PA2	0.866161	0.861	0.861	0.868258	0.721161
Perceptron	0.682194	0.693226	0.693226	0.698548	0.387355
ROMMA	0.873032	0.862548	0.862548	0.870613	0.724
aROMMA	0.873032	0.862548	0.862548	0.870613	0.724
ALMA	0.816871	0.780516	0.780516	0.795645	0.559323
OGD	0.662032	0.679097	0.679097	0.681032	0.359323
SOP	0.939935	0.943548	0.943548	0.946387	0.887097
IELLIP	0.869097	0.862613	0.862613	0.870258	0.724323
CW	0.919613	0.922484	0.922484	0.92629	0.844516
NHERD	0.939677	0.928935	0.928935	0.933323	0.857097
AROW	0.941032	0.940613	0.940613	0.943742	0.880806
NAROW	0.959258	0.892355	0.892355	0.904419	0.789258
SCW	0.918613	0.919968	0.919968	0.923968	0.839484
SCW1	0.936839	0.937516	0.937516	0.940645	0.874516
OurWork	**0.967118**	**0.970156**	**0.970156**	**0.972914**	**0.911014**

Fig. 12. Overall average value of each criteria of network traffic (NSL-KDD data stream).

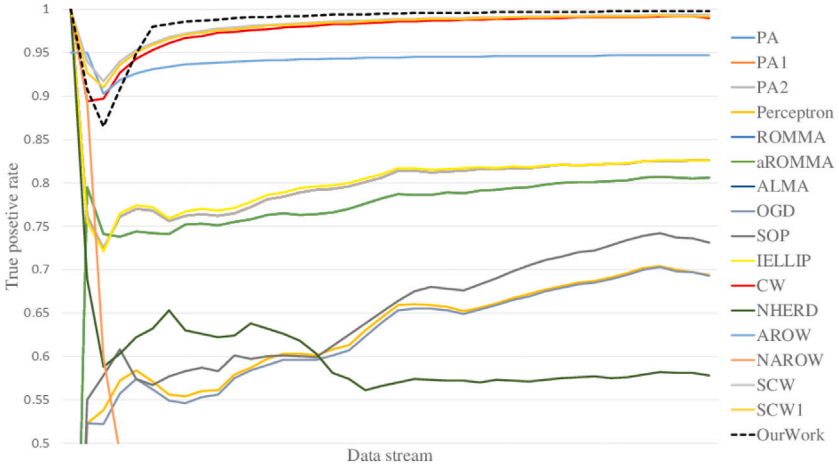

Fig. 13. True positive rate of network traffic (ISCX data stream).

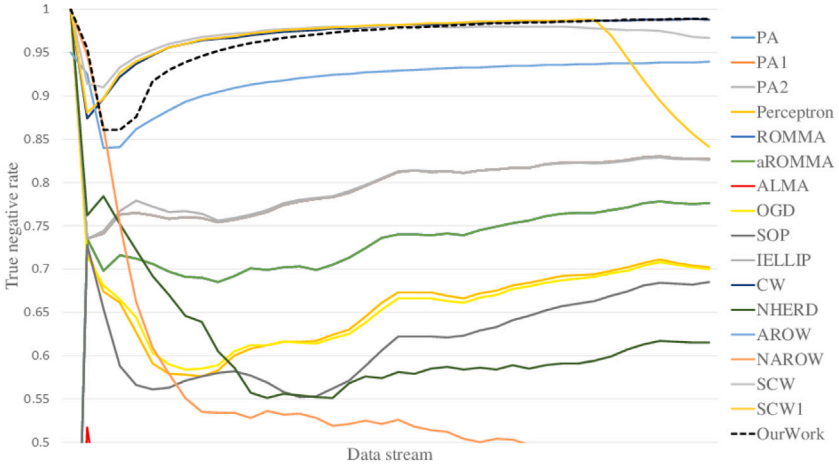

Fig. 14. True negative rate of network traffic (ISCX data stream).

the web page extracted by web parser (Selenium) between April to August 2018. Selenium WebDriver is used to parsing the web page and feature extraction process. For every sample web page, we are using the instance of chrome browser to take the rendering web page for extraction, which is implemented by java. For every URL in the Phishtank and Alexa, we check the web page because sometimes the web page is suspended and cannot access the content.

Electronics data. To evaluate the concept drift in the data stream, many datasets applied. Harries et al. create Electronic data of price sample about Wales Electricity Market, (Harries, 1999). Many of the researchers use this data to the evaluation of concept drift.

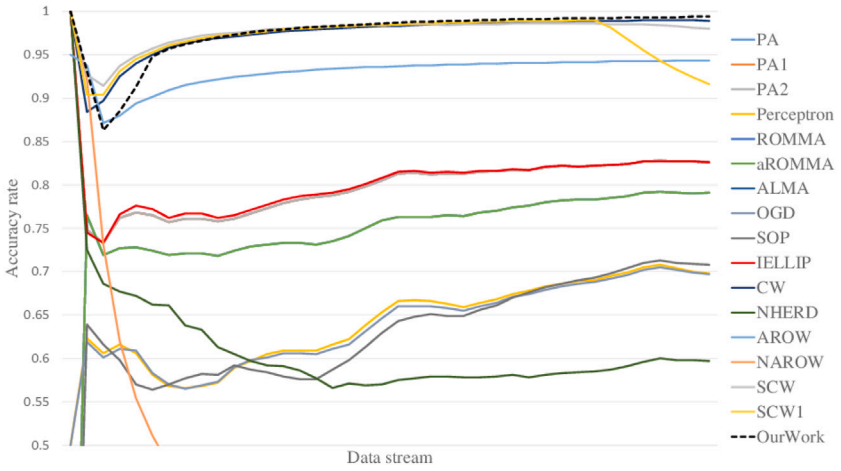

Fig. 15. Accuracy rate of network traffic (ISCX data stream).

	TPR	TNR	Accuracy	Fscore	MCC
PA	0.802325	0.799775	0.80105	0.801325	0.591846
PA1	0.802325	0.799775	0.80105	0.801325	0.591846
PA2	0.802325	0.799775	0.80105	0.801325	0.591846
Perceptron	0.61605	0.6646	0.640375	0.623325	0.288128
ROMMA	0.75905	0.714525	0.736725	0.74215	0.511897
aROMMA	0.75905	0.714525	0.736725	0.74215	0.511897
ALMA	0.165825	0.17425	0.16995	0.166925	-0.65131
OGD	0.611425	0.663625	0.637525	0.619625	0.282513
SOP	0.637375	0.60385	0.6206	0.626125	0.273487
IELLIP	0.804675	0.80135	0.803125	0.8034	0.596179
CW	0.976925	**0.972125**	0.9746	0.974625	0.947795
NHERD	0.6047	0.6201	0.612375	0.60965	0.20541
AROW	0.941806	0.919481	0.93062	0.932321	0.91066
NAROW	0.315225	0.555225	0.4351	0.3535	-0.16518
SCW	0.982075	0.97095	**0.976575**	**0.976675**	**0.951923**
SCW1	0.98035	0.958	0.969125	0.969775	0.937538
OurWork	**0.985485**	0.965615	0.975575	0.9749	0.948308

Fig. 16. Overall average value of each criteria of network traffic (ISCX data stream).

All of the detection procedure implemented on a personal system with 8 GB of memory space, Windows 10 operating system and, a 64-bit 7-core processor. Each data block is evaluated and compared to other methods which the following criteria considered.

- TPR: (Sensitivity) percentage of normal social traffics that are already truly classified.
  $TPR = \frac{TP}{TP+FN}$
- TNR: (Specificity) the percentage of attacks that are correctly determined as abnormal traffics in the social network.
  $TNR = \frac{TN}{TN+FP}$
- Accuracy: The number of data that correctly detected given by all of the data streams.
  $Accuracy = \frac{TN+TP}{TP+FN+TN+FP}$
- F-score: F-measure defined harmonic precision.
  $F-score = \frac{2TP}{2TP+FP+FN}$
- MCC: This method defines the quality of classification by the Matthews Correlation Coefficient model.
  $MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

The evaluation results of the online learners and our work when KDD data stream received by each data block. The Fig. 9 shows that SOP and $AROW$ move without maximal changing and $Perceptron$ is the changeable online learner than the other one. Also among these two learners, $NAROW$ has the highest TPR, but $AROW$ is increasing its TPR over time. The Fig. 10 shows that SOP has the maximum value of TNR and both of the $perceptron$ and OGD had the highest variation. Finally, Fig. 11 shows that between all of the online learner, our work has maximum value (about 97%) of accuracy. Also based on overall averaging of evaluation, our work has the highest rank of criteria. (shown in Fig. 12).

We show another experiment by ISCX dataset. In the first steps, all of the online learning algorithms have the same view of the current data stream. Which their TPR, TNR result change into growth that can cause a positive effect on our work. In this phase that shows in Fig. 13 CW and SCW1 have the maximal value of TPR, TNR. But SCW1, SCW, and CW growth rate are declined over time. As shown in Figs. 13, 14, 15, 16, Given these changes of learner's view, the performance of our work depends on the best of the learner that has the minimum change like SCW with about 98% accuracy.

We investigate the new data set where contain the malicious web page. Our system analyzed based on the current web page. Although compared to the best result (shown in Figs. 17, 18, 19, 20), our system is not reliable than other online learners in the initial times, but the performance has developed over time. As shown in Fig. 19, in the initial steps, SOP is the most efficient detector which has maximum values at around 94% among all above online learner. The accuracy of SOP converges to the constant values, but SCW1, AROW, NERD, and CW is growing over time. The number of the inner URL employed in the page which can be the same or different to the original page. Moreover, these links can be similar properties to malicious web pages when checking their extracted features. In the final steps, we analyze these links, and

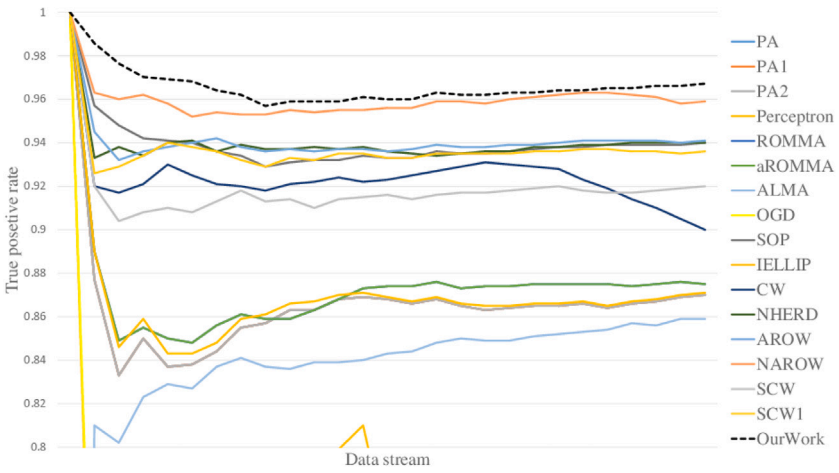

Fig. 17. True positive rate of malicious web page data stream.

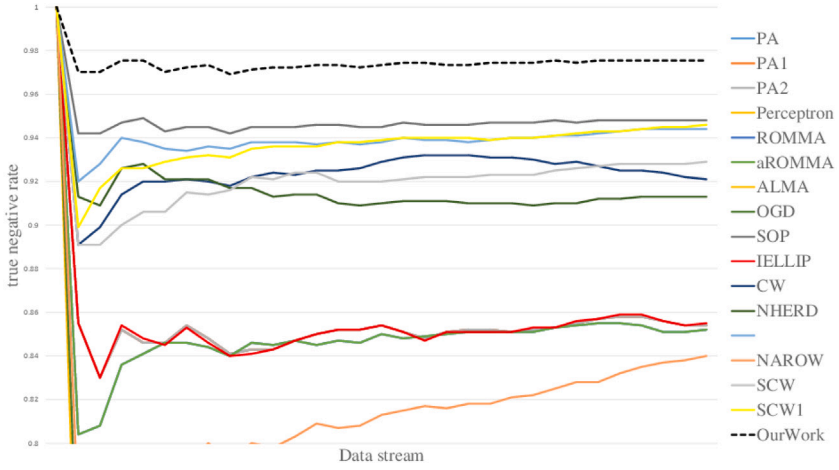

Fig. 18. True negative rate of malicious web page data stream.

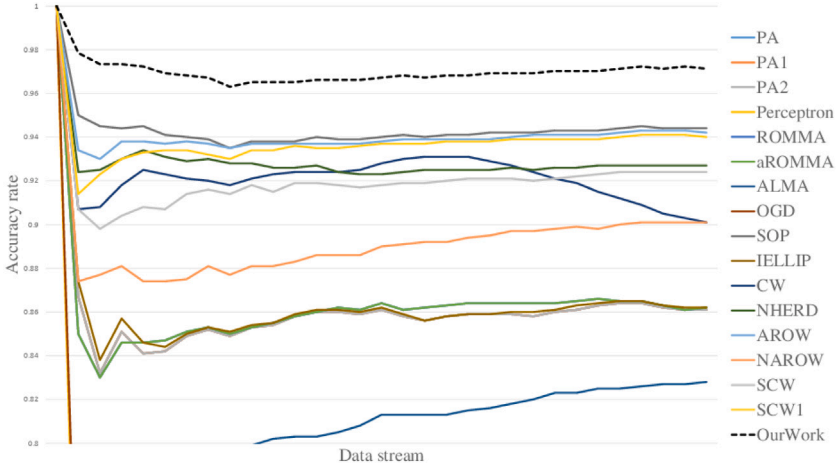

Fig. 19. Accuracy rate of malicious web page data stream.

the result shows our system has a minimum change with about 98% accuracy than other online learning algorithms.

The TPR and TNR values of electronic data stream shows in (Fig. 21), (Fig. 22), respectively. The TNR value for $NAROW$, SOP, and $AROW$ is the highest among other online algorithms, but over time, there is a significant decline in the TNR result of $NHERD$, $AROW$ algorithms. Moreover, some of the linear-order online algorithms such as PA, $PA1$, $PA2$, $Perceptron$, $IELLIP$ have the highest performance between the other learner, because the linear-order is better than Gaussian-order algorithms when the sudden and immediate drift observed in the data stream. So as shown in Fig. 23, the accuracy of the proposed algorithms

	TPR	TNR	Accuracy	Fscore	MCC
PA	0.861	0.855323	0.861	0.934625	0.870063
PA1	0.861	0.855323	0.861	0.934625	0.870063
PA2	0.861	0.855323	0.861	0.934625	0.870063
Perceptron	0.693226	0.705484	0.693226	0.882625	0.891438
ROMMA	0.862548	0.850548	0.862548	0.946625	0.892438
aROMMA	0.862548	0.850548	0.862548	0.946625	0.892438
ALMA	0.780516	0.739452	0.780516	0.915188	0.829313
OGD	0.679097	0.697968	0.679097	0.88275	0.89575
SOP	0.943548	0.947774	0.943548	0.977063	0.954313
IELLIP	0.862613	0.85529	0.862613	0.935375	0.871438
CW	0.922484	0.92571	0.922484	0.96925	0.93825
NHERD	0.928935	0.916484	0.928935	0.969125	0.939375
AROW	0.940613	0.940258	0.940613	0.9805	0.961313
NAROW	0.892355	0.817161	0.892355	0.976563	0.953875
SCW	0.919968	0.921452	0.919968	0.973813	0.947875
SCW1	0.937516	0.937968	0.937516	0.970438	0.940938
OurWork	**0.970156**	**0.9743**	**0.970156**	**0.98925**	**0.987675**

Fig. 20. Overall average value of each criteria of malicious web page data stream.

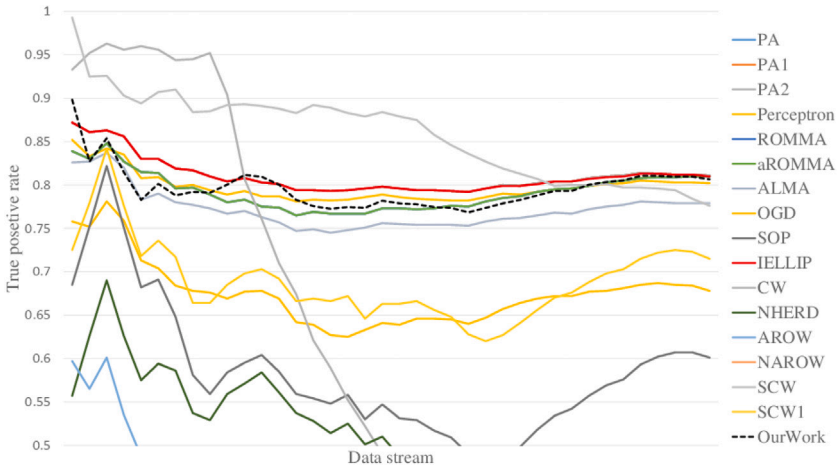

Fig. 21. True positive rate of electronic data stream.

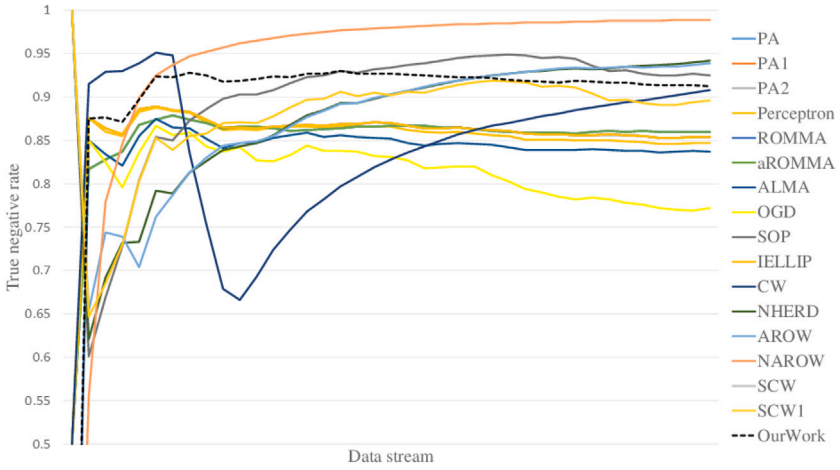

Fig. 22. True negative rate of electronic data stream.

is more than 84% which is better than other online learners (overally shows in Fig. 24).

5. Conclusions

In this paper, we provide an efficient method for discovering the concept drift in the social network data stream. Proposed method depends on Uncertainty Error Correlation Matrix of error rate. Since to tuning detection system for concept drift problem, our method consists two part: online learning algorithms as an expert and the combining of each learner's perspective. Online learning algorithms employed for detecting the type of concept drift such as immediate, gradual, replicating, and development drift by Linear-order and Gaussian-order online learning algorithm. Each of the online learners considers the

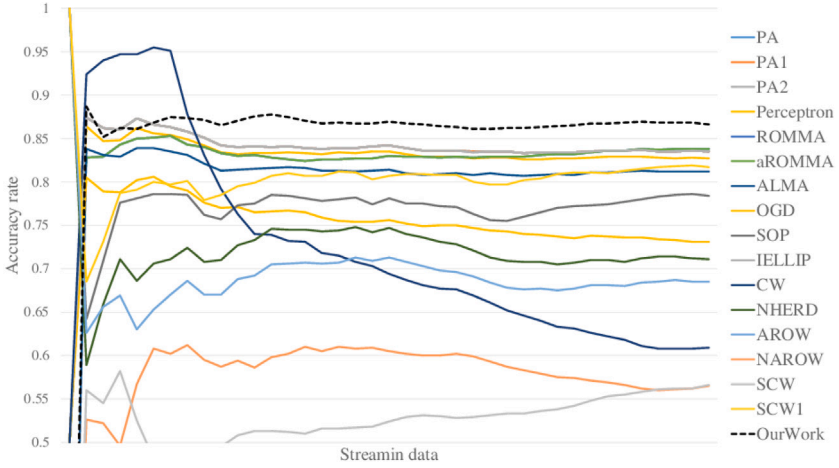

Fig. 23. Accuracy rate of electronic data stream.

	TPR	TNR	Accuracy	Fscore	MCC
PA	0.810737	0.855564	0.833872	0.811132	0.675684
PA1	0.810737	0.855564	0.833872	0.811132	0.675684
PA2	0.810737	0.855538	0.833846	0.811132	0.675658
Perceptron	0.797763	0.851513	0.826231	0.801105	0.659289
ROMMA	0.792974	0.851821	0.824744	0.798684	0.655684
aROMMA	0.792974	0.851821	0.824744	0.798684	0.655684
ALMA	0.772079	0.824718	0.795103	0.777737	0.620026
OGD	0.675184	0.807385	0.751154	0.699263	0.496868
SOP	0.583421	0.902077	0.775718	0.677789	0.527263
IELLIP	0.810553	0.855564	0.833846	0.811132	0.675658
CW	0.557842	0.838436	0.719769	0.595026	0.426211
NHERD	0.493605	0.878077	0.723615	0.590211	0.413053
AROW	0.4155	0.878949	0.692487	0.520447	0.341158
NAROW	0.061921	**0.929769**	0.568	0.096684	0.038553
SCW	**0.858026**	0.284333	0.515744	0.603684	0.178579
SCW1	0.690026	0.878462	0.804821	0.742447	0.585
OurWork	0.796569	0.893406	**0.84526**	**0.827551**	**0.692783**

Fig. 24. Overall average value of each criteria of electronic data stream.

different type of concept drifts in the data stream. In the ensemble phase, we required each online learner's perspective to increase the performance of prediction. Thus, we take a block sequence of a data stream and compute the weight of learner's feedback based on the uncertainty error value of all online learner. There is no data stored in the memory for the prediction where state changes in every block iteration. Detection procedure can be estimated by $O(1)$ overhead without retraining process. The result shows that most of the time, our method access to the highest performance but, in some cases, our method orient to the best online learner. Also, the proposed approach is very practical in the real environment such as the social network.

CRediT authorship contribution statement

Emad mahmodi: Conceptualization, Methodology, Software, Formal analysis, Validation, Investigation, Resources, Writing - original draft, Writing - review & editing, Visualization. **Hadi Sadoghi Yazdi:** Conceptualization, Formal analysis, Data curation, Writing - original draft, Visualization, Project administration. **Abbas Ghaemi Bafghi:** Methodology, Validation, Data curation, Writing - original draft, Visualization, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Amrutkar, C., Kim, Y. S., & Traynor, P. (2017). Detecting mobile malicious webpages in real time. *IEEE Transactions on Mobile Computing*, *16*(8), 2184–2197.

Babcock, B., Babu, S., Datar, M., Motwani, R., & Widom, J. (2002). Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems* (pp. 1–16). ACM.

Baena-García, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavaldà, R., & Morales-Bueno, R. (2006). Early drift detection method.

Bagheri, E., Lu, W., & Ghorbani, A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. In *Second IEEE symposium on computational intelligence for security and defense applications* (pp. 1-6). IEEE. year=2009.

Bifet, A., & Gavalda, R. (2009). Adaptive learning from evolving data streams. In *International symposium on intelligent data analysis* (pp. 249–260). Berlin, Heidelberg: Springer.

Cesa-Bianchi, N., Conconi, A., & Gentile, C. (2005). Second-order perceptron algorithm. *SIAM Journal on Computing*, *34*(3), 640–668.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (1993). Online passive-aggressive algorithms. *Journal of Machine Learning Research (JMLR)*, *1993*, 551–585.

Crammer, K., Kulesza, A., & Dredze, M. (2009). Adaptive regularization of weight vectors. In *Advances in neural information processing systems* (pp. 414–422).

Crammer, K., & Lee, D. D. (2010). Learning via gaussian herding. In *Advances in neural information processing systems* (pp. 451–459).

Deypir, Mahmood, Sadreddini, Mohammad Hadi, & Hashemi, Sattar (2012). Towards a variable size sliding window model for frequent itemset mining over data streams. *Computers & Industrial Engineering*, *63*(1), 161–172.

Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 71–80). ACM.

Dredze, M., Crammer, K., & Pereira, F. (2008). Confidence-weighted linear classification. In *Proceedings of the 25th international conference on machine learning* (pp. 264–271). ACM.

Escovedo, T., Koshiyama, A., da Cruz, A. A., & Vellasco, M. (2018). Detecta: abrupt concept drift detection in non-stationary environments. *Applied Soft Computing*, *62*, 119–133.

Farid, D. M., Zhang, L., Hossain, A., Rahman, C. M., Strachan, R., Sexton, G., & Dahal, K. (2013). An adaptive ensemble classifier for mining concept drifting data streams. *Expert Systems with Applications*, *40*(15), 5895–5906.

Fdez-Riverola, F., Iglesias, E. L., Díaz, F., Méndez, J. R., & Corchado, J. M. (2007). Applying lazy learning algorithms to tackle concept drift in spam filtering. *Expert Systems with Applications*, *33*(1), 36–48.

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In *Brazilian symposium on artificial intelligence* (pp. 286–295). Berlin, Heidelberg: Springer.

Gentile, C. (2001). A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research (JMLR)*, *2*(Dec), 213–242.

Gupta, Brij, B., Tewari, Aakanksha, Jain, Ankit Kumar, & Agrawal, Dharma P. (2017). Fighting against phishing attacks: state of the art and future challenges. *Neural Computing and Applications*, *28*(12), 3629–3654.

Harries, M. (1999). *Splice-2 comparative evaluation: electricity pricing: Tech. rep.*, The University of New South Wales.

Haykin, S. S. (1986). *Adaptive filter theory (vol. 2)*. Englewood Cliffs, NJ: Prentice-hall.

Hsiao, W. F., & Chang, T. M. (2008). An incremental cluster-based approach to spam filtering. *Expert Systems with Applications*, *34*(3), 1599–1608.

Hulten, G., Spencer, L., & Omingos, P. D. (2001). Time-changing data streams. In *7th ACM SIGKDD int. conf. on knowledge discovery and data mining* (pp. 97–106).

Irie, B., & Miyake, S. (1988). Capabilities of three-layered perceptrons. In *IEEE international conference on neural networks (vol. 1, no. 641–648)* (p. 218).

Krawczyk, Bartosz, Minku, Leandro L., Gama, Joao, Stefanowski, Jerzy, & Woźniak, Michał (2017). Ensemble learning for data stream analysis: A survey. *Information Fusion*, *37*, 132–156.

Lane, T., & Brodley, C. E. (1998). Approaches to Online Learning and Concept Drift for User Identification in Computer Security. In *KDD* (pp. 259–263).

Li, Y., & Long, P. M. (2000). The relaxed online maximum margin algorithm. In *Advances in neural information processing systems* (pp. 498–504).

Liang, G., Weller, S. R., Zhao, J., Luo, F., & Dong, Z. Y. (2017). A framework for cyber-topology attacks: Line-switching and new attack scenarios. *IEEE Transactions on Smart Grid*.

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2011). Learning to detect malicious urls. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 30.

Mamun, Mohammad Saiful Islam, Rathore, Mohammad Ahmad, Lashkari, Arash Habibi, Stakhanova, Natalia, & Ghorbani, Ali A. (2016). Detecting malicious URLs using lexical analysis. In *Network and system security* (pp. 467–482). Springer International Publishing.

Mena-Torres, D., & Aguilar-Ruiz, J. S. (2014). A similarity-based approach for data stream classification. *Expert Systems with Applications*, *41*(9), 4224–4234.

Nair, Smrithy Girijakumari Sreekantan, & Balakrishnan, Ramadoss (2018). Mitigating false alarms using accumulator rule and dynamic sliding window in wireless body area. *CSI Transactions on Ict*, *6*(2), 203–208.

Perez-Solano, J. J., & Felici-Castell, S. (2015). Adaptive time window linear regression algorithm for accurate time synchronization in wireless sensor networks. *Ad Hoc Networks*, *24*, 92–108.

Rao, Routhu Srinivasa, & Pais, Alwyn Roshan (2018). Detection of phishing websites using an efficient feature-based machine learning framework. *Neural Computing and Applications*, 1–23.

Ross, G. J., Adams, N. M., Tasoulis, D. K., & Hand, D. J. (2012). Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters*, *33*(2), 191–198.

Somasundaram, Akila, & Reddy, Srinivasulu (2019). Parallel and incremental credit card fraud detection model to handle concept drift and data imbalance. *Neural Computing and Applications*.

Sun, Y., Teng, L., Yin, S., Liu, J., & Li, H. (2017). Study a join query strategy over data stream based on sliding windows. In *International conference on data mining and big data* (pp. 334–342). Cham: Springer.

Tennant, M., Stahl, F., Rana, O., & Gomes, J. B. (2017). Scalable real-time classification of data streams with concept drift. *Future Generation Computer Systems*, *75*, 187–199.

Torquati, Massimo, Mencagli, Gabriele, Drocco, M., Aldinucci, Marco, Matteis, Tiziano De, & Danelutto, Marco (2017). On dynamic memory allocation in sliding-window parallel patterns for streaming analytics. *The Journal of Supercomputing*, 1–18.

Wang, Yongjian, & Li, Hongguang (2018). A novel intelligent modeling framework integrating convolutional neural network with an adaptive time-series window and its application to industrial process operational optimization. *Chemometrics and Intelligent Laboratory Systems*, *179*, 64–72.

Wang, Shuo, Minku, Leandro L., & Yao, Xin (2018). A systematic study of online class imbalance learning with concept drift. *IEEE Transactions on Neural Networks and Learning Systems*, *99*, 1–20.

Wang, J., Zhao, P., & Hoi, S. C. (2012). Exact soft confidence-weighted learning. arXiv preprint arXiv:1206.4612.

Yang, L., Jin, R., & Ye, J. (2009). Online learning by ellipsoid method. In *Proceedings of the 26th annual international conference on machine learning* (pp. 1153–1160). ACM.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning* (pp. 928–936).

E. Mahmodi received his B.Sc. and M.Sc. degrees in Computer Software Engineering from University of Shomal and Ferdowsi in 2012 and 2017, respectively. He is also a member of Data and Communication Security Lab (DCSL) and Pattern Recognition Lab (PRL) in Department of Computer Engineering, Ferdowsi University of Mashhad, Iran. His research interests include Social networking, recommendation system, and Pattern Recognition, Machine Learning, Data Mining to create adaptive system, and securing users against cyber spying and developing advanced authentication schemes for thwarting phishing.

H. Sadoghi Yazdi is currently Professor of Computer Science and Engineering at Ferdowsi University of Mashhad (FUM). He received his B.S. degree in Electrical Engineering from FUM in 1994, and received his M.S. and Ph.D. degrees in Electrical Engineering from Tarbiat Modares University in 1996 and 2005, respectively. His research interests are in the areas of Pattern Recognition, Machine Learning, Machine Vision, Signal Processing, Data Mining and Optimization.

A. Ghaemi-Bafghi received his B.Sc. degree in Applied Mathematics in Computer from Ferdowsi University of Mashhad, Iran, in 1995. He received his M.Sc. and Ph.D. degrees in Computer engineering from Amirkabir (Tehran Polytechnique) University of Technology, Iran in 1997and 2004, respectively. He is member of Computer Society of Iran (CSI) and Iranian Society of Cryptology (ISC). He is an associated professor in Department of Computer Engineering, Ferdowsi University of Mashhad, Iran. His research interests are in cryptology and security, and he has published more than 80 conference and journal papers.