Modern Applied Science; Vol. 16, No. 4; 2022
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

An Online Machine Learning Approach to Sentiment Analysis in Social Media

Jaber Alwidian[1], Tariq Khasawneh[2], Mahmoud Alsahlee[2] & Ali Safia[2]

[1] The Group Securities, Doha, Qatar

[2] Princess Sumaya University for Technology (PSUT), Amman, Jordan

Correspondence: Jaber Alwidian, The Group Securities, Doha, Qatar.

Received: October 5, 2022 Accepted: October 24, 2022 Online Published: October 27, 2022

doi:10.5539/mas.v16n4p29 URL: https://doi.org/10.5539/mas.v16n4p29

Abstract

The online learning, is one that continuously adapts to arriving data, and gets updated incrementally instance by instance. In this paper, we compare the performance of different online machine learning algorithms for the task of sentiment analysis on challenging text datasets. We assess the models using a wide range of metrics, such as microF1, macroF1, accuracy, and running time. Our experiments have revealed that these online models provide a viable alternative to traditional offline machine learning in sentiment analysis, in fraction of the time.

Keywords: online learning, sentiment analysis, real-time text analysis, online machine learning

1. Introduction

The abundance of data has resulted in the rise of many data-driven applications, such applications have contributed to the process of data utilization and its generation simultaneously. The continuous process of data generation has resulted in a huge amount of data containing hidden insights unmined. This type of data, which is huge in its volume, having high variability, and being generated with high speeds from multiple sources is what is now known as Big Data. Additionally, the rise of big data has imposed additional requirements to the process of data handling and storage, resulting in the arrival of new method with high flexibility, speed, and capabilities to cope with these requirements (Li and Manoharan, 2013). In addition to handling and storing this type of data, real-time processing is required to adapt to the velocity in which big data arrives at a predictive system to have a sufficiently fast response or prediction, avoiding expensive delays. To achieve this, online learning has the potential to provide such responsiveness and is being investigated further by experts in the field (Lin and Kolcz, 2012). Some of the applications requiring processing data streams in a real-time manner include the processing of internet logs, sensor networks, user online behavior, and opinion mining in social networks. Generally, a model that is updated using online learning, is one that continuously adapts to arriving data, and gets updated incrementally instance by instance, therefore, it doesn't require "full memory" as in the case of batch learning (Gama et al., 2014; Shahin et al., 2019).

The general architectures of the two approaches, online learning and traditional batch learning is shown in figures 1 and 2.

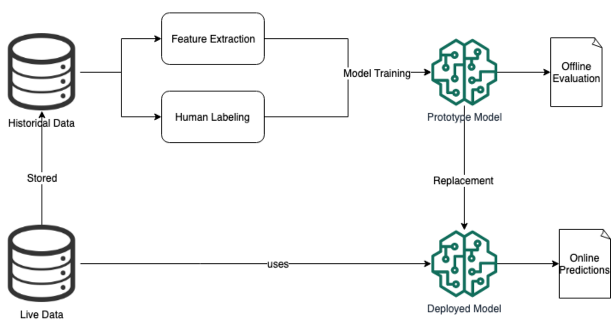

Figure 1. Batch learning

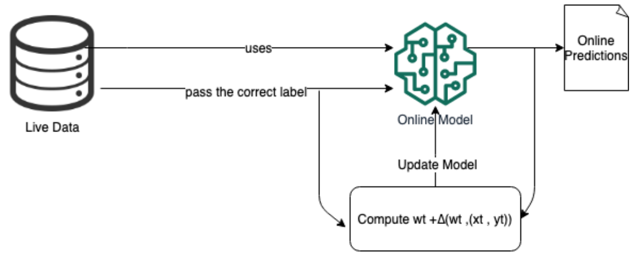

Figure 2. Online learning

Another challenge posed by big data is its variability which means data is being generated in different formats, such as image, video, and text which constitutes the largest portion among them. Therefore, many studies have focused on text analysis. Particularly, sentiment extraction from text, whether it's a binary sentiment classification (e.g. positive or negative), or multi-label classification for a wider range of human emotions. Which motivated this study, to investigate online machine learning for the task of sentiment analysis.

The contributions of this paper can be summarized as follows:

1) Testing the performance of multiple online algorithms for the task of binary sentiment classification.
2) Testing the performance of a wide range of online algorithms for the task of multi-class sentiment classification.
3) The online models providing accuracy comparable to the state of the art of batch models have been demonstrated, achieving 86% on Binary-classification task on the IMBD dataset (Potts, 2010), and 74% on multi-classification task using twitter dataset (Kaggle, 2015).
4) A wide range of metrics for assessing the model has been provided, such as Macro F1, precision, recall, and the average training time for each model.
5) The online neural network has been tested (Sahoo et al., 2017) on both datasets, and it has shown great performance with a trend implying a great predictive potential on larger datasets.

The remainder of this paper is organized as follows. Section 2 provides an overview of related works where online learning is used in mining social networks. In section 3, the different learning methods and how they compare to online learning are introduced. Subsequently, section 4 discusses the setup of the experiment, and the results obtained by analyzing multiple online learning algorithms for the problem of sentiment analysis. In section 5, an interpretation of the results and a conclusion is provided. Finally, the work is concluded in section 6, suggesting potential future work.

2. Literature Review

2.1 Traditional ML for Sentiment Analysis

Many approaches have been proposed to tackle the problem of sentiment analysis, starting with traditional supervised machine learning models, such as the SNoW architecture (Yang and Ahuja, 2001), where classifying the emotional intent of sentences is the objective.

A performance comparison on classifying emotional content using the SNoW learning architecture, a naïve baseline, BOW approach and a tripartite model was done in (Alm, Roth, and Sproat, 2005), pressing a more robust sequential model that considers a larger subset of emotions.

Naturally, two basic problems arise when mining emotions. Firstly, determining the most appropriate emotion for describing a certain text. Secondly, analyzing the prosodic contour to extract emotional content (Cahn, 1990). The second problem is a critical property giving advantage to speech-based emotion mining over text-based prediction tasks. Therefore, justifying the use of text to speech synthesis.

Another proposal was to incorporate syntactic features, resulting from the syntactic connection between words and a given target (Jiang et al., 2011). This is particularly useful in social media content, since coherency is not required, social media posts often lack consistency, and might be comprised of multiple unrelated ideas. Moreover, individual posts might be connected in some way, either through topic or through the same person, such information is of a paramount importance in sentiment classification.

2.2 Deep Learning for Sentiment Analysis

This work has proposed a new model called AdaRNN that enables or integrates the RNN with the target dependent features by finding the words syntactically connected with the interested target as it contains more than one and choose them according to the context and linguistic tags. It also helps the sentiment propagations to be sensitive to both linguistic and semantic categories. Also, a comparison has been done between based line methods of Twitter sentiment classification and the AdaRNN and the AdaRNN has outperform the base line methods (Dong et al., 2014).

2.3 Other Approaches to Sentiment Analysis

2.3.1 Graph Based

Graph-based Sentiment Optimization provides an ideal data structure for encoding sequential models, where context is emphasized for classifying sentiment. In (Jiang et al. 2011 and Salah et al, 2019), three main pillars constitute the context: Retweets, Tweets source, Tweets replying to or being replied to by the tweet of interest. Thus, how all of these elements affect the context of a given text is self-explanatory. However, encoding is done by representing tweets with graph nodes and each context relation with a different edge type. Therefore, the class of a tweet sentiment would be a function of its content and its graph neighbors:

$$p(c|\tau, G) = p(c \vee \tau) \sum_{N(d)} p\big(c\big|N(d)\big)p\big(N(d)\big) \tag{1}$$

C: sentiment label

G: the graph

N(d): assigning sentiment labels to neighbors

T: content of the tweet

2.3.2 CNN

A less direct approach toward classifying sentiment would be the use of Convolutional Neural Networks, by utilizing its effectiveness when dealing with high dimensional data, and a smaller number of model parameters compared to other sequential models, such as LSTMs. Experiments on well-known datasets have shown that employing multiple consecutive convolutional layers is effective and outperforms other deep learning models (Kim and Jeong, 2019).

The context is still preserved when using CNNs, because it takes advantage of convolutional kernels which extracts features automatically inheriting syntactic features implicitly through the convolution operation, which considers spatial locality of a high dimensional instance. It has been shown that retaining more contextual information is possible by using convolutional kernels with attention technique (Gehring et al., 2016; Alwidian et al., 2020). Rios and Kavuluru (2015) have shown that even a single convolutional layer, and a combination of convolutional filters achieves a performance comparable to other state of the art models.

2.3.3 Online Learning for Sentiment Analysis

The preceding approaches to sentiment analysis are all batch-based (offline models), that is the model requires training before being deployed. However, this approach suffers from multiple setbacks in a real- world setting. Firstly, offline models assume the data is available for training beforehand and that it is representative of future instances, thus having expensive memory and time complexity. Secondly, the rigidity of offline models, which means such models are locked for real time modifications, and can’t learn from newly introduced instances, this is usually solved by specifying a time interval in which the model is updated on the new data to capture newly added patterns. Finally, offline models would suffer from bad predictive performance, if real-time data exhibits concept drift, especially if it has a distribution that is considerably different from that of the data in which it was trained on.

Online machine learning methods are able to overcome the offline models drawbacks. Because offline learning models aim to learn the best predictor and continuously apply instant updates to the predictor for each received instance. The online models’ ability to sequentially learn from each instance in a one pass manner. They are suitable for large scale machine learning tasks, and applications involving data of high volume, velocity, and variability. For these reasons, it is assumed that implementing online machine learning in text analysis and opinion mining tasks would demonstrate a predictive performance comparable to that of traditional offline machine learning techniques. Therefore, it is empirically proven to be true in section 5 (Alwidian et al., 2012).

3. Learning Branches

3.1 Incremental Learning

Incremental Learning is a learning scheme in which modification on the model take place incrementally with the arrival of each instance or group of instances (batch). Incorporating incremental learning in traditional machine learning algorithms is becoming the norm, reducing the computational complexity resulting from the learning method being applied to a huge amount of data. In incremental learning, this complexity is avoided by learning from batches of data to the point where the whole data is covered. This approach can be seen in (Cauwenberghs and Poggio, 2000) where training support vector machines has taken a place by using an online recursive algorithm, that considers one vector at a time.

It is worth noting that online learning is a subbranch of incremental learning, where the batch size contributing to the incremental learning process is equal to one, i.e., a single instance. A comparative study between the two incremental approaches can be found in (Read et al., 2012; Alwidian et al., 2018).

3.2 Sequential Learning

Sequential learning is a learning scheme that is mainly concerned with sequential learning. Sequential data is of particular important, because it arises in a variety of applications, such as semantic segmentation (Roscher 2013 and Alwedyan et al, 2011), predicting human behavior (Liciotti et al., 2020), and opinion mining (Lin and Kolcz, 2012). A critical property of sequential data is that the instances are not independent, on the contrary, sequential data exhibits a significant level of sequential correlation.

3.3 Stochastic learning

Stochastic learning algorithms differ from other learning methods in their use of stochastic optimization.

An alternative to batch learning discussed in the context of incremental learning, another commonly used method is stochastic learning, which at each iteration randomly picks an instance to drive the learning process. Significantly reducing the required learning time. Moreover, the use of Stochastic Gradient Descent (SGD) accelerates the optimization speed of a given loss function by looking at a single instance instead of the whole batch. Another common practice which is between the two approaches is the mini-batch SGD (Shalev-Shwartz et al., 2012; Alwidian et al., 2016).

Stochastic gradient descent differs from batch gradient descent as follows:

By simply using a finite training set of independent observations, the following loss function is obtained:

$$\begin{gathered} Traningset: \\ \{(x_1, y_1), \dots\dots, (x_n, y_n)\} C_l(w) \\ = \frac{1}{N} * \sum_{n=1}^{N} L((x, y)_n, w) \end{gathered} \tag{2}$$

Updates differ between the two models:

$$\begin{gathered} forbatch: w_{t+1} = \\ w_t - \propto \frac{1}{N} \frac{\partial L}{\partial w} \sum_{n=1}^{N} L((x, y)_n, w) \end{gathered} \tag{3}$$

Instead of averaging the gradient of the loss over the complete training set, each iteration of the online gradient descent consists of choosing an example $(x, y)_i$ at random and updating the parameter w according to the following formula.

$forSGD$:

$$\begin{gathered} w_{t+1} = w_t - \propto \frac{\partial L}{\partial w} L((x, y)_i, w_i), \\ where\ i\ is\ randomly\ chosen, \\ i \in [1, N] \end{gathered} \tag{4}$$

3.4 Adaptive Learning

This learning schema is used whenever the model operates in a continuously evolving environment, in which adaptation is required to maintain a sufficient performance. However, this adaptation doesn't necessarily imply that the model is online, because adaptive leaning doesn't have to follow online learning theory, it can employ a heuristic approach for the adaptation process. A comprehensive overview of the applications of supervised adaptive learning on data streams is provided by (Carpenter, Grossberg, and Reynolds, 1991; Al-Fayoumi et al., 2020). However, it should be noted that adaptive learning has expanded to include other learning branches such as reinforcement learning, which is inherently an adaptation problem. It should be noted that online learning and adaptive learning have a somewhat fuzzy borders separating the two learning branches, since adapting to an evolving environment usually requires a responsive learning mechanism, like the one used in online learning algorithms.

3.5 Reinforcement Learning

Reinforcement learning is the training of machine learning models to make a sequence of decisions. The objective of the agent is to find a close to optimal policy in a complex, evolving environment, attempting to maximize its cumulative reward. A critical difference between Reinforcement learning and traditional supervised learning, is that instead of approximating a mapping between the input and the output, the objective in RL is to find an input (policy x) that would maximize the reward R(x). It should be noted that reinforcement learning is closely related to online multi armed bandit, but RL encompasses a wider range of problems, where the environment might be more complex than that in the multi armed bandit problems (Alsahlee et al., 2019).

3.6 Interactive Learning

It is a branch of learning were the domain expertise is involved in the learning process itself. in contrast, traditional machine learning approaches expect the data collected to be prepared, cleaned and enhanced by domain experts if necessary. In addition to involving the expert in the learning loop, an enhancement to interactive learning can be implemented, were queries posed by the active classifier to the user are provided with an explanation to the user on the motive behind such query. Improving the model's predictive and explanatory powers (Teso and Kersting, 2019; Abuqabita et al., 2019).

4. Online Learning Architectures

To implement an online learning pipeline in a big data setting, two main architectures can be utilized:

4.1 Lambda Architecture

It is a data processing pipeline that is used to serving the environments that have a huge amount of data in an efficient, robust way, with regards to latency reduction, it provides a fault-tolerant and robust environment. It is an appropriate example of using a batch learning pipeline.

Lambda architecture consists of three layers:

• Batch Layer

It acts like the persistence layer; the batch layer receives data from the data source store it in whatever database then doing on it some operation such as batch predictions or some processing using map-reduce functionality. The lambda architectural pattern can be seen in figure 3 (Hadi et al., 2010).

Batch Layer can be used as a layer for batch learning or offline learning to make the training against the whole data including historical data.

• Speed Layer (Streaming layer)

It is used as a volatile layer for latency reduction and for serving queries in real-time or near real-time. Speed Layer can host the ML model that is created by the Batch Layer, for predicting online.

• Serving Layer

It is a view layer for the Batch view and Real-time view.

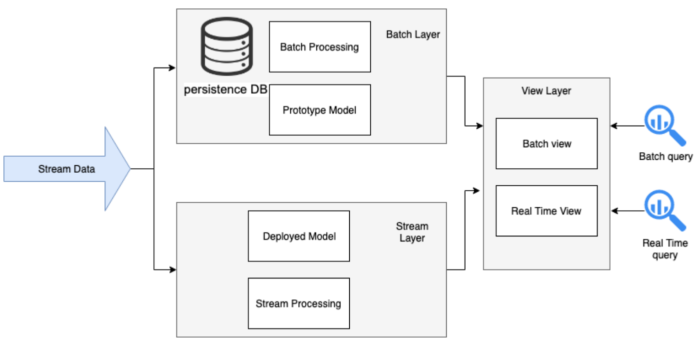

Figure 3. Lambda

4.2 Kappa Architecture

Kappa architecture (shown in figure 4) only has a streaming layer, so it is the best place to use online learning because there is no persistence layer, and the historical data does not exist or stored only for reporting manner.

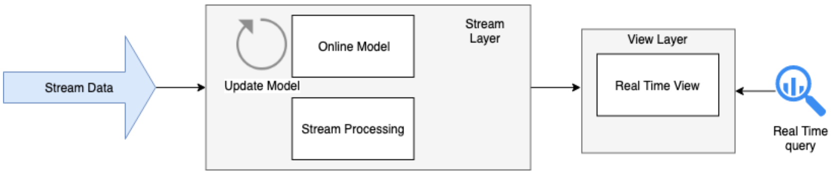

Figure 4. Kappa

5. Experiment

5.1 Hardware Setup

The hardware has been used in this experiment to drive the following results has the following specifications: Intel Core i7-8750H CPU-2.2GHz, Graphic card RTX 2070 with MAX-Q, RAM 32GB.

Text classification pipeline

Data gathering and preprocessing:

By preprocessing we mean the action of applying some or all of the following:

1) Cleaning the text from noisy characters and patterns.
2) Removing of punctuation.
3) Stemming.
4) Dropping stop words or the commonly used characters from text.

Text numerical representation (vectorization):

There are a lot of techniques to encode the sentences into a representative set of features. Text vectorization for text classification task is representing the corpus numerically to prepare it for fitting by the ML model. The sent2vec (Pagliardini, Gupta, and Jaggi, 2017) technique was used in this experiment for expanding the dimensionality to 700 dimensions.

5.2 Sent2vec

Sent2vec is based on FastText, a library, this library is powered by Facebook research team. It operates on the level of characters, improving the performance on sentences with rare word occurrences. In FastText, each word is represented by a bag of characters (n-grams), which takes in consideration the subwords information. This technique is powerful in making representative conversion for the short texts with rare terms. Additionally, it's convenient to use for vectorization of streaming text, because all sentences would be converted to a fixed number of dimensions (Bojanowski et al., 2017; Abdallat et al., 2019).

5.3 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a technique to compute the relevancy of terms found in a document to a corpus of documents.

Generally, TFIDF is comprised of two main stages (Qaiser and Ali, 2018); In stage 1, Term Frequency (TF) measures the term frequency in the received document. While in stage 2, Inverse Document Frequency (IDF) measures how the term is frequent against the whole corpus. Therefore, the final equation will be as follow:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \tag{5}$$

Where:

$tf_{i,j}$: is the term frequency in the document level.

N: frequency of the term among the whole corpus.

df: the number of the documents in the corpus.

Model fitting and tuning.

Finally, the model is used for fitting, and is tuned through learning.

IMDB: Binary Classification [Potts and Christopher, 2011; Moh'd Iqbal et al., 2013]

This data set contains binary movie reviews, the main data set contains 50,000 reviews labeled with 0 for positive sentiment and 1 for negative sentiment. This dataset is balanced by having 25k of each class. Figures 5,6, and 7 show the performance of different online classifiers on this dataset, where the performance has been measured using the accuracy (fig. 5), the Macro F1 (fig. 6 and fig. 7). It should be noted that figure 7 is the results obtained after excluding 25% of the incoming instances, this was done in order to display the performance of the classifiers after dismissing the contribution of the initial 25% of the instances, where the classifier had a poor performance. It can be seen that after excluding this effect, the most algorithms provide a stable performance with impressive Macro F1 which scores of 86%, 85%,86%,83% for the ASGD, logistic regression, ONN, and the bagging classifier respectively.

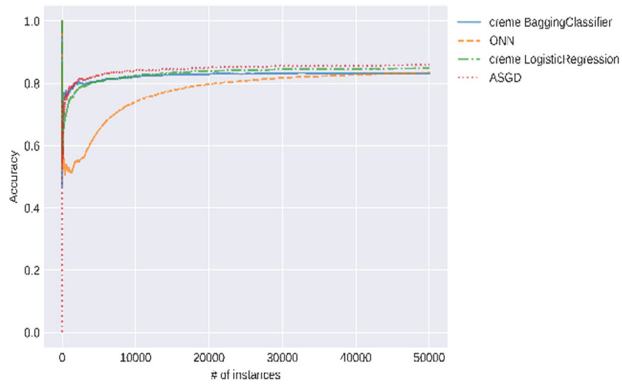

Figure 5. IMDB Accuracy for the algorithms

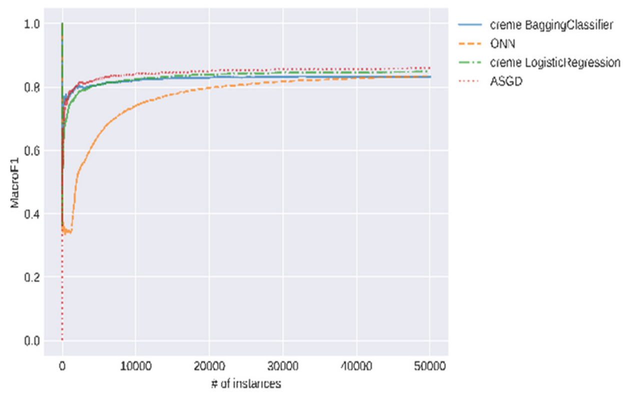

Figure 6. IMDB MacroF1 for the algorithms

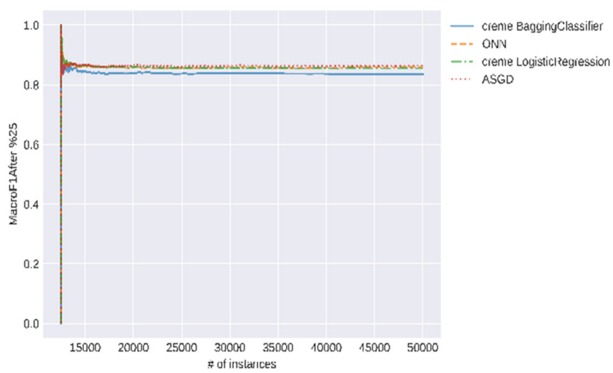

Figure 7. IMDB MacroF1 for the algorithms after 2

An overview of the algorithms performance using multiple metrics is shown in table 1, displaying the results obtained on the IMDB dataset.

Table 1. Evaluation Criteria For The Algorithms Over Imdb

ALLGORITHM	MACROF1	PRECISION	RECALL	ACCURACY	AVG TIME	MACROF1AFTER25%
CREME BAGGINGCLASSIFIER	0.83214	0.83038	0.8348	0.83214	0.01238	0.83498
ONN	0.83404	0.83514	0.8324	0.83404	0.00502	0.85868
CREME LOGISTICREGRESSION	0.84796	0.8471	0.8492	0.84796	0.00202	0.85471
ASGD	0.85921	0.8574	0.86176	0.85921	0.0003	0.86485

5.4 Twitter: (Multi-Class) [Kaggle, 2015 and Fasha et al., 2020]

This data set contains 3 class for the emotional expressions, the main data set contains 14,427 reviews labeled with 0 for positive sentiment, 1 negative sentiment, and 2 for neutral. Thus, the negative sentiment represents 63%, and 21% for neutral sentiment, and 16% for positive sentiment. Figures 8, 9, and 10 display the results for the accuracy, and MacroF1 excluding the first 25% of the instances, which were obtained by the GaussianNB, ONN, ASGD, perceptron, and passive aggressive classifiers on the twitter dataset.

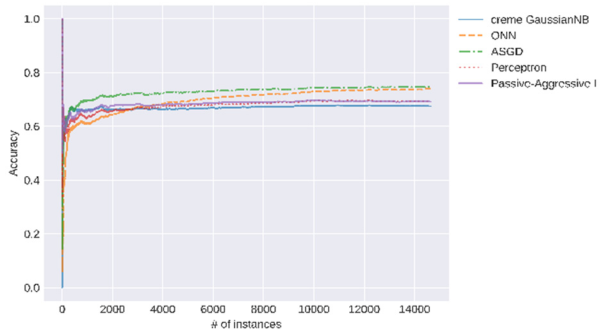

Figure 8. Twitter Accuracy for the algorithms

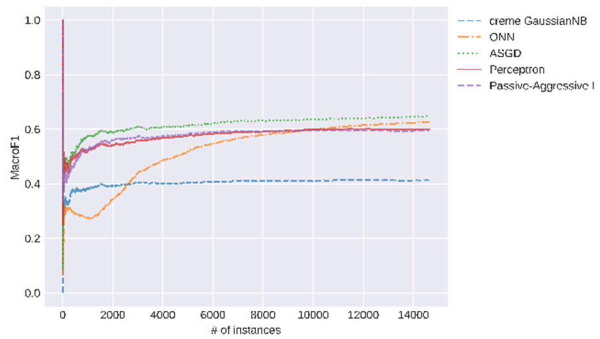

Figure 9. Twitter MacroF1 for the algorithms

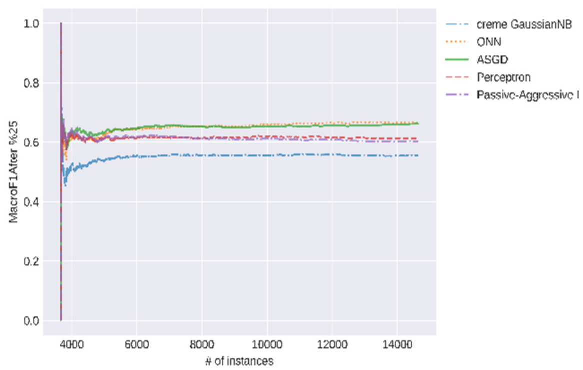

Figure 10. Twitter MacroF1 for the algorithms after 25%

Table 2 provides the results obtained by different classifiers on the twitter dataset, where the performance of each algorithm was measured using different metrics, such as MacroF1, MicroF1, accuracy, etc.

Table 2. Evaluation Criteria For The Algorithms Over Twitter

ALLGORITHM	MACROF1	MICROF1	PRECISION	RECALL	ACCURACY	AVG TIME	MACROF1AFTER25%
CREME GAUSSIANNB	0.41223	0.67439	0.45307	0.40047	0.67439	0.00651	0.55413
ONN	0.62688	0.739	0.68587	0.59809	0.739	0.00492	0.66746
ASGD	0.6476	0.74751	0.69338	0.62157	0.74751	0.00125	0.66129
PERCEPTRON	0.59983	0.69374	0.60367	0.59635	0.69374	0.00133	0.61202
PASSIVE-AGGRESSIVE I	0.59382	0.69156	0.60066	0.58806	0.69156	0.00119	0.60179

6. Discussion

Simple pre-processing techniques such as removing noisy characters, tokenization and stemming were used in this research, the preprocessed text is then vectorized by the sent2vec.

The demonstrated models are ASGD, ONN, Logistic Regression and bagging classifier, which were evaluated using a variety of metrics, such as accuracy, microF1, macroF1, etc.

The results have shown that the algorithms have similar results, except for the ONN which was inferior to other algorithms, however, the ONNs trend showed a potential for a better performance on large-scale datasets. Especially after excluding the first 25% of the incoming instances, reducing the penalty on the accumulated accuracy due to the poor initial performance.

7. Conclusion and Future Work

There are a number of interesting avenues for future work. Other studies have included analysis of online algorithms' resilience to concept drift, including an evaluation of data streams mining (Mittal and Kashyap, n.d.), techniques for concept drift detection (Patil, 2019), and the utilization of online ensemble classifiers to learn from non-stationary data streams (Verdecia-Cabrera, Blanco, & Carvalho, 2018). Additionally, the study of a multiclass online classifiers would be important, since customer's behavior is often non-binary. Replacing the embedding stage with a TF-IDF stage and study the performance difference would carry additional useful insights. Lastly, a full deployment of the online model in a big data environment would help validate the results further in a real-time world scenario.

References

Abdallat, A. A., Alahmad, A. I., & AlWidian, J. A. (2019). Hadoop mapreduce job scheduling algorithms survey and use cases. *Modern Applied Science, 13*(7), 38-48. https://doi.org/10.5539/mas.v13n7p38

Abuqabita, F., Al-Omoush, R., & Alwidian, J. (2019). A comparative study on big data analytics frameworks, data resources and challenges. *Modern Applied Science, 13*(7), 1-14. https://doi.org/10.5539/mas.v13n7p1

Al-Fayoumi, M., Alwidian, J., & Abusaif, M. (2020). Intelligent association classification technique for phishing website detection. *The International Arab Journal of Information Technology, 17*(4), 488-496. https://doi.org/10.34028/iajit/17/4/7

Alsahlee, O. H., Al-Zu'bi, A., Alamro, I. I., & Alwidian, J. (2019, July). Distributed and automated machine learning in big data stream analytics. In *2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4)* (pp. 307-313). IEEE. https://doi.org/10.1109/WorldS4.2019.8903932

Alwedyan, J., Hadi, W. E. M., Salam, M. A., & Mansour, H. Y. (2011, April). Categorize arabic data sets using multi-class classification based on association rule approach. In *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications* (pp. 1-8). https://doi.org/10.1145/1980822.1980840

Alwidian, J., Abu-Mansour, H., & Ali, M. (2012, March). Efficient algorithm for two dimensional pattern matching problem (non-square pattern). In *2012 International Conference on Information Technology and e-Services* (pp. 1-8). IEEE. https://doi.org/10.1109/ICITeS.2012.6216622

Alwidian, J., Hammo, B. H., & Obeid, N. (2018). WCBA: Weighted classification based on association rules algorithm for breast cancer disease. *Applied Soft Computing, 62*, 536-549. https://doi.org/10.1016/j.asoc.2017.11.013

Alwidian, J., Hammo, B., & Obeid, N. (2016). Enhanced CBA algorithm based on apriori optimization and statistical ranking measure. In *Proceeding of 28th international business information management association (IBIMA) conference on vision* (Vol. 2020, pp. 4291-4306).

Alwidian, J., Rahman, S. A., Gnaim, M., & Al-Taharwah, F. (2020). Big Data Ingestion and Preparation Tools. *Modern Applied Science, 14*(9), 12-27. https://doi.org/10.5539/mas.v14n9p12

Anthony, R., & Kavuluru, R. (2015). Convolutional Neural Networks for Biomedical Text Classification: Application in Indexing Biomedical Articles. In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, 258-267. https://doi.org/10.1145/2808719.2808746

Cahn, J. E. (1990). The Generation of Affect in Synthesized Speech. *Journal of the American Voice I/O Society, 8*(1), 1.

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991). ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network. *Neural networks, 4*(5), 565-588. https://doi.org/10.1016/0893-6080(91)90012-T

Daniele, L., Bernardini, M., Romeo, L., & Frontoni, E. (2020). A Sequential Deep Learning Application for Recognising Human Activities in Smart Homes. *Neurocomputing, 396*, 501-513. https://doi.org/10.1016/j.neucom.2018.10.104

Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., & Xu, K. (2014). Adaptive Recursive Neural Network for Target-Dependent Twitter Sentiment Classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers), 49-54. https://doi.org/10.3115/v1/P14-2009

Doyen, S., Pham, Q., Lu, J., & Hoi, S. C. H. (2017). Online Deep Learning: Learning Deep Neural Networks on the Fly. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence Main track*, 2660-2666. https://doi.org/10.24963/ijcai.2018/369

Fasha, M., Hammo, B., Obeid, N., & Widian, J. (2020). A hybrid deep learning model for arabic text recognition. *International Journal of Advanced Computer Science and Applications, 11*(8), 122-130. https://doi.org/10.14569/IJACSA.2020.0110816

Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A Survey on Concept Drift Adaptation. *ACM computing surveys (CSUR), 46*(4), 1-37. https://doi.org/10.1145/2523813

Gert, C., & Poggio, T. (2000). Incremental and Decremental Support Vector Machine Learning. *Advances in neural information processing systems, 13*.

Hadi, W. E. M., Salam, M. A., & Al-Widian, J. A. (2010, June). Performance of NB and SVM classifiers in Islamic Arabic data. In *Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications* (pp. 1-6). https://doi.org/10.1145/1874590.1874604

Hannah, K., & Jeong, Y.-S. (2019). Sentiment Classification Using Convolutional Neural Networks. *Applied Sciences, 9*(11), 2347. https://doi.org/10.3390/app9112347

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). Target-Dependent Twitter Sentiment Classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 151-160.

Jimmy, L, & Kolcz, A. (2012). Large-Scale Machine Learning at Twitter. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 793-804.

Jonas, G., Auli, M., Grangier, D., & Dauphin, Y. N. (2016). *A Convolutional Encoder Model for Neural Machine Translation.*

Li, Y., & Manoharan, S. (2013). A Performance Comparison of SQL and NoSQL Databases. In *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing* (PACRIM), 15-19. https://doi.org/10.1109/PACRIM.2013.6625441

Matteo, P., Gupta, P., & Jaggi, M. (2017). *Unsupervised Learning of Sentence Embeddings Using Compositional N-Gram Features*.

Mittal, V., & Kashyap, I. (2018). Performance Evaluation and Estimation for Concept Drifting Data Stream Mining. *International Journal of Computer Applications, 180*(42), 10-15. https://doi.org/10.5120/ijca2018917105

Moh'd Iqbal, A. L., Hadi, W. E., & Alwedyan, J. (2013). Detecting phishing websites using associative classification. *Journal of Information Engineering and Applications, 3*.

Ovesdotter, A, C., Roth, D., & Sproat, R. (2005). Emotions from Text: Machine Learning for Text-Based Emotion Prediction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 579-586.

Patil, M. M. (2019). Handling Concept Drift in Data Streams by Using Drift Detection Methods. In Balas, V., Sharma, N., & Chakrabarti, A. (Eds.), *Data Management, Analytics and Innovation* (pp. 155-166). https://doi.org/10.1007/978-981-13-1274-8_12

Piotr, B., Grave, E. Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the association for computational linguistics, 5*, 135-146. https://doi.org/10.1162/tacl_a_00051

Potts, C. (2011). On the Negativity of Negation. In *Proceedings of SALT 20*, pp. 636-659. https://doi.org/10.3765/salt.v20i0.2565

Qaiser, S., & Ali, R. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. *International Journal of Computer Applications, 181*(1), 25-29. https://doi.org/10.5120/ijca2018917395

Read, J., Bifet, A., Pfahringer, B., & Holmes, G. (2012). Batch-Incremental versus Instance-Incremental Learning in Dynamic and Evolving Data. In Hollmén, J., Klawonn, F., & Tucker, A. (Eds.), *Advances in Intelligent Data Analysis XI* (pp. 313-323). https://doi.org/10.1007/978-3-642-34156-4_29

Roscher, R. (2012). *Sequential Learning Using Incremental Import Vector Machines for Semantic Segmentation*.

Salah, H., Al-Omari, I., Alwidian, J., Al-Hamadin, R., & Tawalbeh, T. (2019). Data streams curation for better machine learning functionality and result to serve IoT and other applications: a survey. *Journal of Computer Science, 15*(10), 1572-1584. https://doi.org/10.3844/jcssp.2019.1572.1584

Shahin, D., Hannen Ennab, R. S., & Alwidian, J. (2019). Big Data Platform Privacy and Security: A Review. *IJCSNS, 19*(5), 24.

Shai, S.-S. (2011). Online Learning and Online Convex Optimization. *Foundations and Trends® in Machine Learning, 4*(2), 107-194. https://doi.org/10.1561/2200000018

Stefano, T., & Kersting, K. (2019). Explanatory Interactive Machine Learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 239-245. https://doi.org/10.1145/3306618.3314293

Verdecia-Cabrera, A., Blanco, I. F., & Carvalho, A. C. P. L. F. (2018). An Online Adaptive Classifier Ensemble for Mining Non-Stationary Data Streams. *Intelligent Data Analysis, 22*(4), 787-806. https://doi.org/10.3233/IDA-173522

Yang, M.-H., & Ahuja, N. (2001). Learning to Detect Faces with Snow. In *Face Detection and Gesture Recognition for Human-Computer Interaction*, 123-150. https://doi.org/10.1007/978-1-4615-1423-7_6