An Ensemble Model to Minimize Fluctuation Influences on Short-Term Medical Workload Prediction

Tasquia Mizan[1]

[1]Department of Mechanical and Industrial Engineering,

Ryerson University, 350 Victoria St., Toronto, ON, Canada M5B 2K3

Email: tmizan@ryerson.ca; Tel : +16472038690; Mobile : +16475325128

Sharareh Taghipour[*]

[*]Department of Mechanical and Industrial Engineering,

Ryerson University, 350 Victoria St., Office-EPH 300, Toronto, ON, Canada M5B 2K3

[*]Email: sharareh@ryerson.ca; Tel : 416-979-5000 ext. 7693; Mobile : +16477806708

Abstract

Real-time data are commonly prone to errors due to irregular fluctuations, seasonal biases, and missing values in the data. The erroneous data causes inaccurate forecasting which leads to business loss. Moreover, the concept drift problem is a known problem in time series forecasting that also results in poor forecasting accuracy. The execution time of a machine learning model is also crucial when it is deployed in a real-time environment. This work presents an Adaptive Batched-Ranked Ensemble (ABRE) model that reduces the effect of fluctuation using the time-variant windowing technique. A data aggregation technique is developed and integrated with the offline training phase of the proposed model to tackle the concept drift problem. A meta-model is developed in the online forecasting phase which ensures faster execution for incoming data. The model is implemented for the medical workload prediction after testing and comparing with a few other heterogeneous ensemble models. The comparison results show in terms of the root mean squared error, the proposed model performs at least 65.7% better than the heterogeneous stacked ensemble models on the experimental dataset. Moreover, in comparison to the other standalone models considered in this experiment, the ABRE model reduces the prediction error by approximately 73.6%.

Keywords: Time-series forecasting, Irregular fluctuation, Concept drift, Batched-ranked, Multivariate forecasting model, Ensemble modeling.

1. Introduction

The prediction accuracy of time series data is crucial in almost every industry since inaccurate forecasting leads to business loss. Real-time data is erroneous, for which maintaining high forecasting accuracy is a challenge [1][2]. Machine Learning (ML) forecasting models can tackle this challenge and provide higher forecasting accuracy [3]. As different ML models have their own benefits and drawbacks, researchers sought to combine them for forecasting, which brings forth the idea of ensemble modeling [4]. Multiple ML algorithms are combined to develop an ensemble model. It is proved that the prediction accuracy of an ensemble model for a complex dataset is much higher than a standalone model [5]. This ensemble technique uses a meta-learning stage which ensures the highest accuracy [4] [6]. The parallelization of multiple algorithms at the same time ensures faster training time [7]. For that reason, the ensemble modeling method is widely adopted for predictions on online streaming data [8].

Predictive models for time series forecasting are trained using historical data. In real-world applications, data is online or streamed, i.e., data is available over time. In common cases, the statistical properties of the online data also change over time which makes the data non-stationary [9]. This type of data can have non-linear trends, multi-seasonality, and irregular fluctuations [10]. Fluctuations in the data may occur due to the noise or due to the concept drift [11] [12]. The noise in the data refers to irregular fluctuations [13]. On the other hand, concept drift is a term used in time series forecasting to indicate how the statistical parameters and underlying relationships of the input and target variables change over time [14]. For example, patients' waiting times may change over time depending on the workload distribution of a health care system.

The prediction accuracy of online time-series data may suffer from irregular fluctuations and concept drift, which makes the prediction model obsolete [15]. Learning in non-stationary environments where data is available over time, may change the underlying relationships between the input and output variables [12]. In the ML domain, the problem of changing relationships over time is known as concept drift [14]. Prediction on online time-series data requires faster execution time [16]. Models with faster execution time may result in poor prediction accuracy [17]. The use of an ensemble method can balance this tradeoff between execution time and accuracy [18]. In this research, we have proposed an ensemble model to predict incoming time-series data. Online or streaming time-series data usually exhibit time-varying behaviors, which may result in large variations between old and incoming data causing concept drift [19]. To overcome the problem, a new incremental learning method is applied in this study by continuously retraining the proposed model after a fixed time interval. However, retraining an ensemble model may increase the computational costs [20]. Besides, irregular fluctuations in time-series data may result in poor prediction accuracy [21]. To minimize the retraining cost and to ensure faster execution time, a hybrid offline training and online prediction framework is implemented in this research. Also, a new time-variant batch window method is proposed in this study to handle the effect of irregular fluctuation in data.

Thus, the main contributions of this research are as follows.

- Developing a new time-variant training window technique to denigrate the influence of irregular fluctuations on prediction accuracy.
- Implementing a new method for aggregation of incoming data with the existing trained data to tackle the effect of concept drift.
- Implementing new incremental learning using an averaging approach to prevent data from growing over time.
- Implementing a modified batch ranked method in ensemble modelling to boost up the training phase.
- Implementing a hybrid online and offline prediction framework for prediction to ensure faster execution with the offline retraining phase.

This paper is organized as follows:

Related research on ensemble modelling design and implementations of ML algorithms about short-term time series has been described in Section 2 of this paper. Section 3 describes the proposed method. In Section 4, a short description of the working dataset has been stated. Models' evaluation and key findings have been discussed in Section 5. Section 6 concluded the applicability of the model.

2. Literature review

The ensemble method is an advanced ML technique that combines several algorithms for prediction. Inside the ensemble model, each ML algorithm performs prediction based on its capacity and strengths. Hence, most often, the outcome of an ensemble model outperforms a single model's prediction [21]. In recent years, ensemble models have been widely used in different industries across different problem domains such as forecasting, classification, regression, error reduction, and feature selection. Predictions based on streaming data are commonly performed using ensemble models since a single model may result in high prediction errors for streaming data [7]. Ensembling of multiple models provides higher accuracy, plus the parallelization ensures lower execution time [21]. Ensemble modeling can be based on any of the three hypotheses namely bagging, boosting, and stacking. A bagging method lowers the model's variance [22]. A boosting method decreases the model's bias [23], and stacking enhances the model's predictive power with fine accuracy [24]. In this research, a staking method is used for the prediction of time-series data.

Anifowse et. al. [6] proposed an ensemble model using Artificial Neural Network (ANN) to predict the porosity and permeability of petroleum reservoirs. The experimental result shows the proposed model outperformed Random Forest (RF) model and ensemble model with traditional bootstrap aggregation method. Later, the same authors [25] proposed an ensemble model of Extreme Learning Machine. The experimental result of that study indicated the ensemble models perform better than individual algorithms used as the base learners. A modified version [26] of the proposed model demonstrated the ANN ensemble model with increased search space can provide a higher performance of the model. However, these proposed models did not address the concept drift problem.

Several approaches have been proposed to handle the concept drift depending on how quickly the drift is detected and adopted. Depending on the concept drift handling approach, ensemble models can be further classified into passive and active [27]. In *Active* approaches, when a drift is detected, typically new predictive models are created to learn the new concept and thereby help the system recover from the drift problem [28]. *Passive* approaches maintain an ensemble of *base models* for prediction. Weights are assigned to the models based on their prediction errors. These weights indicate the best corresponding model for the current concept [29]. Ensemble-based approaches dealing with concept drift are categorized into homogeneous and heterogeneous. Active homogeneous approaches are based on single learners combined with a Drift Detection Method (DDM). A threshold point is assigned for drift detection, and when this point is reached the model recognizes it as the occurrence of the concept drift. A common way to learn the new concept is to reset the single learner [30]. Several methods have been suggested for drift detection. DDM is a statistical process for detecting concept drift [31]. It measures an online-learning model's probability of producing minimal error over time and its associated standard deviation. Early Drift Detection Method (EDDM) [32] is similar to DDM but calculates the distance between two misclassifications instead of the error rate. The Statistical Test in Equal Proportion measures the concept drift by comparing the accuracy among the latest examples and the total accuracy from the start of the learning [33]. However, this produces many false positives near the drift points [34]. Passive learning methods do not depend on drift detection techniques, instead, they maintain an ensemble of learners. When a concept drift occurs, a new learner is created to learn the new concept and an older learner is discarded from the ensemble due to its poor performance. Established passive ensembles are implementing dynamic model selection approaches for the inclusion or exclusion of base learners. Most of the heterogeneous ensemble techniques that exist for online data rely on meta-learning [8] which helps to determine which learning techniques are more appropriate for the data.

Ensemble modeling with stacking is extensively used for the prediction of streaming data [35]. An ensemble model with a hybrid incremental learning approach is proposed for short-term electric load forecasting to overcome the effects of drift [17]. Zhu et al. [36] have proposed an ensemble model with parallel incremental learning to bypass the effect of concept drift. As incremental learning results in the continuous growth of data, an ensemble model with a forgetting factor is proposed by Yu and Webb [12] which keeps the data within a specified size and reduces the model’s complexity. Ren et al. [37] have proposed a selection-based resampling ensemble model for faster execution on imbalanced online data. An incremental regression framework using an ensemble model is developed [38] to lower the gradual drifting effect on prediction. Li et al. [14] have proposed a dynamically updated ensemble for learning imbalanced data streams with concept drift. A rank-based ensemble is proposed [16] to predict short-term online data which allows lower runtime of the ensemble model. Similar to the ranked-based approach, a heterogeneous dynamic weighted majority [20] method is also applied to the ensemble modeling. Ancy and Paulraj [39] have used dynamic sampling on their proposed ensemble model to handle imbalanced data with concept drift. A mixture of neural network and support vector machine-based ensemble methods using transfer learning and incremental learning is developed [21] to lower the effects of concept drift. A schema-enabled knowledge graph embedding method is proposed [40] to address the concept drift challenge in the ML domain. Also, a statistical drift detection method is proposed [15] where data distribution is tracked to tackle the concept drift problem. However, the integration of these drift detection methods with the ensemble model makes the model more complex.

Ahmad et al. [41] proposed a hybrid ensemble model to forecast crude oil prices. They combined Group Method Data Handling (GMDH) with the median ensemble empirical mode decomposition to achieve higher forecasting accuracy. However, the execution time of GMDH method is high for the large dataset. Also, the concept drift problem is not addressed in their study. Yang et al. [42] proposed a concept drift tolerant transfer learning technique using the ensemble model. The learners are assigned a weight based on their prediction accuracy and then the higher weightage learners are selected. The learners are updated based on the adaptive weighted correlation alignment. However, this model can perform well when the input attributes are highly correlated. Priya and Uthra [19] also proposed an ensemble model to handle the concept drift problem in streaming data. They used the ‘Majority weighted minority oversampling’ technique to address the imbalanced class problem, and the ‘Diversity for dealing with drifts’ method to deal with the concept drift problem. Their result shows the ensemble model can provide high accuracy; however, the model's execution time is high.

In our proposed model, rank based learner selection approach is used to reduce the weight assignment complexity. The batch aggregation and training method are also implemented to handle the concept drift problem. Finally, the offline training and online prediction with the meta-model is proposed to reduce the model’s execution time. From the studied literature, it is noticed that only a few research considered multivariate regression problems on streaming data. It is also observed that the fluctuation reduction technique is not addressed in the previous literature where multivariate-streaming data is used. As best to our knowledge, the concept drift handling technique with incremental learning is not addressed in the previous literature where multivariate regression problem on streaming data is considered. Table 1 lists the recent research studies based on ensemble modeling and provides a comparison of the models' capacity with our proposed model.

3. Methodology

In this section, the methodology of the Heterogeneous Stack Ensemble (HSE) model is first described. Then the methodology of the proposed Adaptive Batch-Ranked Ensemble (ABRE) model is mentioned. The HSE model is used as a benchmark model to compare the proposed ABRE model's performance.

3.1 Heterogeneous stack ensemble (HSE) model

We have used the HSE model as our benchmark model where several predictive algorithms are stacked. These predictive algorithms are commonly termed the base learners. The predictions of these base learners on the training data, where the target variable is known, produce an error matrix. The formation of this matrix is based on the k-fold cross-validation. Each column of this matrix is the prediction error of individual base learners. After training all base learners, the corresponding values of the target variables are combined to form the final matrix, which is known as the training metadata. The base learners are then exposed to the incoming test data where the target variables are unknown. Each base learner performs prediction on the test data and the values of the prediction are stored as a column to produce another error matrix termed input metadata. A logistic regression model [43] is then trained on the training metadata and performs the prediction on the input metadata. Figure 1 shows the working methodology of the HSE model.

Instead of using the same type of base learners (e.g., collection of RF learners), the predictive learners with different characteristics and prediction capabilities are chosen to make the model robust. The learners used to develop the HSE model are RF, ANN, Support Vector Regression (SVR), Gradient Boosting Machine (GBM), Extreme Gradient Boosting (XGBoost), Multilayer Perceptron (MLP), and Generalized Linear Model (GLM). GLM is the most common technique for regression which is easy to implement and interpret. RF is one of the best supervised-learning algorithms containing the capabilities of providing reliable feature engineering, and working on fewer data points. ANN can handle complex data efficiently. GBM is built for higher accuracy by generating complex tree structures. XGBoost is highly scalable which provides faster execution. MLP is a powerful supervised learning algorithm for regression and multidimensional mapping, which can efficiently handle nonlinear data. The learners' chosen for the HSE model contains different hyperparameters which are listed in Table 2.

3.2 Adaptive Batch-Ranked Ensemble model

In the proposed ABRE model, the offline training method is used to train the model. A set of expert learners are selected during the training and are grouped as an integrated model. An offset of this trained model is made available online to predict the incoming data. At the training phase, the available data is broken down into multiple time windows $[bs_n\ , where\ n = 0\ to\ (n - 1)]$ of 15 minutes. Available data in each window is trained with different ML algorithms known as base learners, $L^{1...l}$.Based on the performance of the base learners, the highest performed learner is selected as the expert learner for that specific time window. This process is carried out for every available time window (bs_n) on the dataset to identify a set of expert learners ($E^{1...M}$)along with the weights (w_j). These expert learners with corresponding weights are passed to the online prediction phase as an integrated model. The incoming test window of the next 15 minutes is fed to that model and the predictions of the expert learners are combined using the weighted average technique. Figure 2 shows the block diagrams of the proposed ABRE model.

The online data windows are aggregated with the main training data after accumulating 96 data windows, or in other words after every (*96*15minutes = 1440 minutes*) 24 hours. The offline and online phases of the model are described in the next parts of this section.

3.2.1 Offline phase

The offline phase mainly focuses on the model's training. The time series prediction model commonly suffers from the concept-drift problem. Retraining the model at a fixed interval is required to tackle this problem [14]. Retraining the ensemble model increases the model's computational costs and results in the delay prediction. In the proposed ABRE model, the offline phase is used to retrain the model so that it does not affect the prediction run time. The training phase is mainly incorporated with two approaches: i) construction of the time-variant training window, and ii) the selection of the expert learners. These two approaches are described in the following subsections.

i. Time variant training window

The available time-series data is divided into 15 minutes time intervals that provide 96 data windows for any given day, and approximately 35,040 windows for a given year (or 35,136 windows for leap year). If the number of input features is denoted as *d*, the input domain for each window becomes $X_t \in R^{s_t \times d_t}; \forall t = 1...96$. The available instances in each time window are denoted as $s^i = \left\{ x_1^i, x_2^i, ..., x_d^i \right\}$, where $s^i \in X_t$ and $1 \le i \le s$. The data instances of the same time window with the same days and the same months, but different years are aggregated together. For the numeric target variable, the values of the target variable are averaged for the same input label. If the input labels are not matched, the sample is added as an instance on that specific window. As an example, the data of January first, 2019 and January first, 2020 have 96 windows each. If window one for both dates has the same input labels, the numeric target variables are averaged and for the different labels, it adds as an instance on window one of January first. The incoming data is also added with these fixed-sized training windows after every 24 hours. To add the incoming test window, we first match the window numbers of the corresponding day and month. If there is no match, the training dataset is updated with the incoming test windows' window numbers, days, months, and corresponding instances. If there is an exact match found for the window number, day, and month, we then check the input features of instances for both data. If input features are exactly matched, the known target variable of the incoming test instances are averaged with the existing training instances. Figure 3 shows the process of the time-variant window.

The idea of averaging the incoming input space has several benefits. One of them is maintaining the input space with a minimum number of training instances which lowers the model's training time [21]. Averaging of target variables also minimizes the irregular fluctuations in the time series data that helps to avoid the concept drift [44].

ii. Selection of expert learners

A set of learning algorithms $L^{1...l}$ are used in the training phase, where $l \geq 0$. The experimental dataset is divided into multiple windows based on time as discussed in the above section. Instead of the conventional approach of dividing the whole dataset into training and testing instances, each window is further divided into 80% of training and 20% of test data. In the conventional approach, ML algorithms commonly use the cross-validation (CV) technique with a specified K-fold to gain higher prediction accuracy. Wherein, the dataset is divided into K number of sets and each set is used to validate the trained model. This process is complex and computationally costly [3] [45]. To reduce this complexity in this study, 20% of each time window is assigned as test data. Each window is transferred through the learning algorithms set L^l. Each learning algorithm termed as the base learner is trained using 80% of the window data, and is ranked r_j using Equation (1) based on their performances on the 20% test portion of the data.

$$r_j = \sum_{bs=0}^{n-1} r_L$$

(1)

where $\forall L = 1...l and (j = 1...M) \in L$

Thus, each algorithm in L^l gets a rank for each window. Root Mean Squared Error (RMSE) is used to determine the performances of these base learners. A learner, which produces the minimum error for a specific window is attained rank one and is considered as an expert learner (E^M)for that window. This ranking process carries on throughout the training process for all time- dependent windows. The total rank (R_M)for expert learners is calculated using Equation (2).

$$R_M = \sum_{j=1}^{M} r_j$$

(2)

After training all windows (bs_n), the rank one learners are grouped to develop the ensemble of expert models. The weight (w_j) of each expert learner is determined by the total number of times an expert learner attains rank one using Equation (3).

$$w_j = \frac{r_j}{R_M}$$

(3)

The selected expert learners are then passed to the online phase to perform predictions on the incoming test data.

3.2.2 Online phase

The selected expert learners from the offline phase are grouped and work as a meta-model in the online phase. The predictions of individual expert learners are combined and averaged using the meta-model. The rank-based expert learner selection process is performed in the offline phase as it requires higher training time and memory space. Only the meta-model is deployed in the online phase to reduce the computational complexity.

The learners (E^M) in this meta-model individually forecast on the incoming 15 minutes time window. These forecasted outputs (y_M) are multiplied with the corresponding weights (w_j)of the expert learners. The resulting numbers are summed up and divided by the total weight (W_M) calculated using Equation (4), which gives us a single prediction output provided in Equation (5).

$$W_M = \sum_{j=1}^{M} w_j$$

(4)

$$F(y) = \frac{1}{W_M}\left[\sum_{1}^{t}\left(w_j \times y_M\right)\right]$$

(5)

The weighted average technique allows the model to forecast with higher accuracy. There are multiple advantages to using this weighted average method over other available techniques, such as boosting or bagging. First, it is simple to implement which reduces the computational costs.

Second, it captures the relative importance of the experts from the training phase, and these important factors indicate which expert learner's predictions can get more focus to ensure better prediction results [46]. Third, it can save the model from overfitting while keeping the prediction variance optimal. The final prediction of the meta-model along with the incoming test data are stored temporarily in the online data windows database. These data are appended to the original training database after accumulating the 96 windows of the online incoming test data and corresponding online predictions.

4. Description of dataset

The proposed model of this study is developed by focusing on a dataset containing radiologists' activities. The radiologist dataset is a multivariate time series data, which is sourced from the IntelePACS server. The proposed ABRE model is a part of an integrated monitoring system of all treatment phases along with the efficient workload distribution in each phase. The target variable in this data is the radiologists' workload. The prediction of the radiologists' workload in advance can allocate the optimal time for their workload, which results in increased productivity. The dataset contains 22 input features and one target variable with 579,799 instances, spread over a two years time horizon. The input features are the mixture of numeric (such as patients' arrival timestamp, date, hours, minutes, etc.) and categorical variables (such as order area, order priority, order modalities, etc.). The target variable contains the numeric values; thus, the developed model is performing regression. The performance of the ABRE model is also tested with ten publicly available datasets listed in Section 5.

5. Experimental results

In the experiment, first, the HSE model is used to predict the workload from the radiologists' data divided into 80% training and 20% testing sets. The model training time with k-fold CV is recorded. The model is tested with the test data and the performance is recorded. The HSE model is then used to predict the workload for the next 15 minutes in a recurrent fashion based on the incoming data.

The ABRE model is also trained with the available 35,136 time-variant windows. Each window contains x_s input instances, where s is the number of instances. The instances in each window are further divided into 80% of training and 20% of testing data, where $s \geq 5$. The windows containing $s \leq 5$ instances are not considered in the training phase. Each window is passed sequentially to the base learners L^l, and each base learner learns from the 80% training instances. Based on their learning, the base learners make predictions on 20% of the testing instances, and their predictions are ranked according to their RMSE. The learner with the lowest RMSE gets rank-1. All rank-1 learners for 35,136 are grouped to form the ensemble of experts (E^M). Table 2 shows the highest rank (r_j) assignment to the base learners for each window to identify the expert learners along with their weight (w_j).

The base learners used in this study with different hyperparameters are shown in the first column of Table 2. The second column shows the number of times a base learner has attained rank-1. We have only four expert learners for the initial training of the ABRE model. These four expert learners are stacked, and an offset of this stacked model is made available online for prediction. To understand the prediction's strength of ensemble modeling for regression, we have recorded the individual performance of the expert learners of the ABRE model. These results are compared with the HSE model and the ABRE model on the experimental dataset, which are shown in Table 3.

From Table 3, it is observed that the ABRE model's execution time is slightly higher than the SVR, because of the ranking process. However, compared to the error measure, the ABRE model outperforms the others. The ABRE model lowers the RMSE by approximately 65.86% and the MASE by approximately 75.84% in comparison with the HSE model. Among the standalone models, XGBoost performed better than RF, SVR, and MLP. However, the average RMSE of the ABRE model is 86.1% less than the XGBoost. Figure 4 shows the comparative performance results of the models using a two-factor Analysis of variance (ANOVA) test (Zimmerman and Zumbo, 1993).

It is noticed from the ANOVA test that the ensemble technique works better than RF, SVR, MLP, and XGBoost for the Radiologist dataset. Further, we have tested the ABRE model with ten publicly available time-series datasets and compared the performances with the HSE model.

To provide the streaming data characteristics on these public datasets, the timestamps are mutated into years, months, days, hours, and minutes. The ‘minutes’ attribute is grouped into 15 minutes time windows and the last 96 windows are considered as the incoming data with the unknown value of the target variables. The comparative results are shown in Table 4.

From the experiment on the public datasets, it is observed that for high volume complex datasets the runtimes of the ABRE model improve more than the HSE model. As an example, the ABRE decreases the runtime by 21% for the “ICU” dataset which has 7931 instances with 14 features. Whereas, it decreases the runtime by 57.7% for the power system data which has 5,02,373 instances with 19 features. For the percentage of improvement in RMSE, the ABRE model works better than the HSE model. However, the HSE model performed better for the traffic data of Brazil which has only 135 instances with 18 attributes. The reason is the lack of data in the ABRE model’s time-variant data windows, which have impacted the training phase of the ABRE model, and thus, produce erroneous results.

6. Conclusion

A nonlinear multivariate time-series regression model to deal with online streaming data is proposed in this study. The proposed model is developed based on the medicare scenario where the radiologists’ workload is predicted for 15 minutes in advance. This prediction is used in another existing model for better scheduling, which is out of scope in this study and has not been described. Prediction accuracy plays a critical role in such a system where a slight increase in prediction accuracy boosts up profits. Ensemble technique by combining multiple predictive algorithms is a proven method that produces better predictions compared to a single model. The proposed ABRE model incorporates this ensemble technique to achieve better accuracy in prediction. Sometimes this higher prediction accuracy may result in a model’s overfit. In this research, the prediction horizon is 15 minutes which is a short-term prediction. So, the overfitting problem can be overlooked. The HSE model used in this study is a traditional stack ensemble model, which is also used as a benchmark model to be compared with the proposed ABRE model. A meta-learning stage is used to make the HSE model more powerful. Execution time is also an important factor in predicting online streaming data where a lower execution time is required. A CV technique provides higher model accuracy, but it is computationally costly. A rank-based model selection technique is implemented in the proposed ABRE method instead of the traditional CV technique to lower the execution time. The experimental results of this study show the ABRE model provides 48.27% faster execution than the HSE model, when the power system dataset with 60 minutes intervals is applied. For the power system data with 30 minutes intervals, the ABRE model has provided a 54.67% faster runtime in comparison to the HSE model. The proposed ABRE model has produced even faster execution, approximately 57.71% faster than the HSE model, when the power system data with 15 minutes intervals is used. This result suggests the ABRE model's execution performance increases for a shorter time period. Most often, time-series data suffer from gradual concept drift problems which can be either incremental or decremental. Incremental learning is a proven method to handle concept drift [21]. The ABRE model uses the same technique of incremental learning by adding the incoming 15 minutes data to the training dataset and retraining the model after every 96 data windows. The prediction accuracy of a time-series model may fall short because of irregular fluctuation in training data. The ABRE model uses an aggregated instances technique for each unique batch of data to handle the fluctuation problem. Moreover, the fixed number of batches for training data also ensures the lower retraining time of the model.

Real-time data is complex which may consist of high dimensionality combined with errors in data. The dataset used in this study also has a very large number of records combined with a large number of input features. Some parts of the data are inputted by humans, and the other parts are from machine generation. Data errors are common in such datasets. The experimental results show that the ABRE model performs better in such complex scenarios. The proposed model is tested on several publicly available datasets, and the result is compared with our base model's (HSE) prediction on those datasets. Based on the results, the following conclusions are made:

1. The substitution of the CV process with ranking lowers the model execution time.
2. The time-variant training window learning is beneficial to minimize the influences of irregular fluctuations for short-term time-series forecasting.
3. The proposed ABRE approach outperforms single structure models. The ABRE also provides the desired accuracy without the meta-learning stage described in the HSE model.

4. The offline retraining phase eliminates the effects of concept drift on time series modeling.

It is noticed that if the number of training instances is too low, the HSE model performs better than the ABRE model, which is the observed limitation of the proposed ABRE model. The ABRE model is developed for single target prediction where we are predicting the workload of the Radiologists. Developing this model for multi-target predictions is our future scope.

Acknowledgment

Funding: This study was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) (EGP 524631-2018). The authors would like to thank NSERC and the Government of Canada for their support.

Availability of data and material: The data that supports the findings of this study was made available by Intelerad Medical Systems as part of NSERC (EGP524631-2018) collaboration, and its availability is subject to a contractual obligation.

References

[1] Chen, K., & Huang, R. “Robust empirical likelihood for time series”, *Journal of Time Series Analysis*, 42(1), 4-18 (2021).

[2] Imani, S., Madrid, F., Ding, W., et al. “Introducing time series snippets: a new primitive for summarizing long time series”, *Data Mining and Knowledge Discovery*, 34(6), 1713-1743 (2020).

[3] Lu, H. J., Zou, N., Jacobs, R., et al. “Error assessment and optimal cross-validation approaches in machine learning applied to impurity diffusion”, *Computational Materials Science*, 169, 109075 (2019).

[4] Xi, J., & Yamauchi, H. “A Layer-wise Ensemble Technique for Binary Neural Network”, *International Journal of Pattern Recognition and Artificial Intelligence* 2152011 (2021).

[5] Birman, Y., Hindi, S., Katz, G., et al. “Cost-effective ensemble models selection using deep reinforcement learning”, *Information Fusion*, 77, 133-148 (2022).

[6] Anifowose, F., Labadin, J., & Abdulraheem, A. “Ensemble learning model for petroleum reservoir characterization: a case of feed-forward back-propagation neural networks”, *In Pacific-Asia*

Conference on Knowledge Discovery and Data Mining, Springer, Berlin, Heidelberg, April, pp. 71-82 (2013).

[7] van Rijn, J. N., Holmes, G., Pfahringer, B., et al. "The online performance estimation framework: heterogeneous ensemble learning for data streams", *Machine Learning*, 107(1), 149-176 (2018).

[8] Jomaa, H. S., Schmidt-Thieme, L., & Grabocka, J. "Dataset2vec: Learning dataset meta-features", *Data Mining and Knowledge Discovery*, 35(3), 964-985 (2021).

[9] Vishwakarma, G. K., Paul, C., & Elsawah, A. M. "A hybrid feedforward neural network algorithm for detecting outliers in non-stationary multivariate time series", *Expert Systems with Applications*, 184, 115545 (2021).

[10] Shi, Z., Bai, Y., Jin, X., et al. "Parallel deep prediction with covariance intersection fusion on non-stationary time series", *Knowledge-Based Systems*, 211, 106523 (2021).

[11] Neto, A. F., & Canuto, A. M. "EOCD: An ensemble optimization approach for concept drift applications", *Information Sciences*, 561, 81-100 (2021).

[12] Yu, H., & Webb, G. I. "Adaptive online extreme learning machine by regulating forgetting factor by concept drift map", *Neurocomputing*, 343, 141-153 (2019).

[13] Zhou, J., Sun, M., Han, D., et al. "Analysis of oil price fluctuation under the influence of crude oil stocks and US dollar index—Based on time series network model", *Physica A: Statistical Mechanics and its Applications*, 582, 126218 (2021).

[14] Li, Z., Huang, W., Xiong, Y., et al. "Incremental learning imbalanced data streams with concept drift: The dynamic updated ensemble algorithm", *Knowledge-Based Systems*, 195, 105694 (2020).

[15] Micevska, S., Awad, A., & Sakr, S. "SDDM: an interpretable statistical concept drift detection method for data streams", *Journal of Intelligent Information Systems*, 1-26 (2021).

[16] Mizan, T., & Taghipour, S. "A causal model for short- term time series analysis to predict incoming Medicare workload", *Journal of Forecasting*, 40(2), 228-242 (2021).

[17] Qiu, X., Suganthan, P. N., & Amaratunga, G. A. "Ensemble incremental learning random vector functional link network for short-term electric load forecasting", *Knowledge-Based Systems*, 145, 182-196 (2018).

[18] Ayachi, R., Afif, M., Said, Y., et al. "Real-time implementation of traffic signs detection and identification application on graphics processing units", *International Journal of Pattern Recognition and Artificial Intelligence*, 35(07), 2150024 (2021).

[19] Priya, S., & Uthra, R. A. "Comprehensive analysis for class imbalance data with concept drift using ensemble based classification", *Journal of Ambient Intelligence and Humanized Computing*, 12(5), 4943-4956 (2021).

[20] Idrees, M. M., Minku, L. L., Stahl, F., et al. “A heterogeneous online learning ensemble for non-stationary environments”, *Knowledge-Based Systems*, 188, 104983 (2020).

[21] Wang, J., Mo, Z., Zhang, H., et al. “Ensemble diagnosis method based on transfer learning and incremental learning towards mechanical big data”, *Measurement*, 155, 107517 (2020).

[22] Cassales, G., Gomes, H., Bifet, A., et al. “Improving the performance of bagging ensembles for data streams through mini-batching”, *Information Sciences*, 580, pp.260-282 (2021).

[23] Trizoglou, P., Liu, X., & Lin, Z. “Fault detection by an ensemble framework of Extreme Gradient Boosting (XGBoost) in the operation of offshore wind turbines”, *Renewable Energy*, 179, 945-962 (2021).

[24] Radhakrishnan, P., Ramaiyan, K., Vinayagam, A., et al. “A stacking ensemble classification model for detection and classification of power quality disturbances in PV integrated power network”, *Measurement*, 175, 109025 (2021).

[25] Anifowose, F. A., Labadin, J., & Abdulraheem, A. “Ensemble model of non-linear feature selection-based extreme learning machine for improved natural gas reservoir characterization”, *Journal of Natural Gas Science and Engineering*, 26, 1561-1572 (2015).

[26] Anifowose, F., Labadin, J., & Abdulraheem, A. “Towards an improved ensemble learning model of artificial neural networks: lessons learned on using randomized numbers of hidden neurons”, *In Artificial Intelligence: Concepts, Methodologies, Tools, and Applications*, *IGI Global*, pp. 325-356 (2017).

[27] Neto, A. F., & Canuto, A. M. “EOCD: An ensemble optimization approach for concept drift applications”, *Information Sciences*, 561, 81-100 (2021).

[28] Guo, H., Zhang, S., & Wang, W. “Selective ensemble-based online adaptive deep neural networks for streaming data with concept drift”, *Neural Networks*, 142, 437-456 (2021).

[29] Zheng, X., Li, P., Hu, X., et al. “Semi-supervised classification on data streams with recurring concept drift and concept evolution”, *Knowledge-Based Systems*, 215, 106749 (2021).

[30] Liu, F., Qian, Y., Li, H., et al. “CAFFNet: Channel Attention and Feature Fusion Network for Multi-target Traffic Sign Detection”, *International Journal of Pattern Recognition and Artificial Intelligence*. 35(07), 2152008 (2021).

[31] Mahdi, O. A., Pardede, E., Ali, N., et al. “Diversity measure as a new drift detection method in data streaming”, *Knowledge-Based Systems*, 191, 105227 (2020).

[32] Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, et al. “Early drift detection method”, *In Fourth international workshop on knowledge discovery from data streams*, Vol. 6, pp. 77-86 (2006).

[33] Mejri, D., Limam, M., & Weihs, C., “A new time adjusting control limits chart for concept drift detection”, *IFAC Journal of Systems and Control*, 17, 100170 (2021).

[34] de Barros, R. S. M., Hidalgo, J. I. G., & de Lima Cabral, D. R. “Wilcoxon rank sum test drift detector”, *Neurocomputing*, 275, 1954-1963 (2018).

[35] Sun, J., Fujita, H., Chen, P., et al. “Dynamic financial distress prediction with concept drift based on time weighting combined with Adaboost support vector machine ensemble”, *Knowledge-Based Systems*, 120, 4-14 (2017).

[36] Zhu, L., Ikeda, K., Pang, S., et al. “Merging weighted SVMs for parallel incremental learning”, *Neural Networks*, 100, 25-38 (2018).

[37] Ren, S., Zhu, W., Liao, B., et al. “Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning”, *Knowledge-Based Systems*, 163, 705-722 (2019).

[38] Xiao, J., Xiao, Z., Wang, D., et al. “Short-term traffic volume prediction by ensemble learning in concept drifting environments”, *Knowledge-Based Systems*, 164, 213-225 (2019).

[39] Ancy, S., & Paulraj, D. “Handling imbalanced data with concept drift by applying dynamic sampling and ensemble classification model”, *Computer Communications*, 153, 553-560 (2020).

[40] Chen, J., Lécué, F., Pan, J. Z., et al. “Knowledge graph embeddings for dealing with concept drift in machine learning”, *Journal of Web Semantics*, 67, 100625 (2021).

[41] Ahmad, W., Aamir, M., Khalil, U., et al. “A New Approach for Forecasting Crude Oil Prices Using Median Ensemble Empirical Mode Decomposition and Group Method of Data Handling”, *Mathematical Problems in Engineering 2021*, pp.1-12 (2021).

[42] Yang, C., Cheung, Y. M., Ding, J., et al. “Concept Drift-Tolerant Transfer Learning in Dynamic Environments”, *IEEE Transactions on Neural Networks and Learning Systems*, (2021).

[43] Cox, D. R. “The regression analysis of binary sequences”, *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2), 215-232 (1958).

[44] Gu, B., Quan, X., Gu, Y., et al. “Chunk incremental learning for cost-sensitive hinge loss support vector machine”, *Pattern Recognition*, 83, 196-208 (2018).

[45] Xiong, Z., Cui, Y., Liu, Z., et al. “Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation”, *Computational Materials Science*, 171, 109203 (2020).

[46] Werbin-Ofir, H., Dery, L., & Shmueli, E. “Beyond majority: Label ranking ensembles based on voting rules”, *Expert Systems with Applications*, 136, 50-61 (2019).

List of captions

Figures:

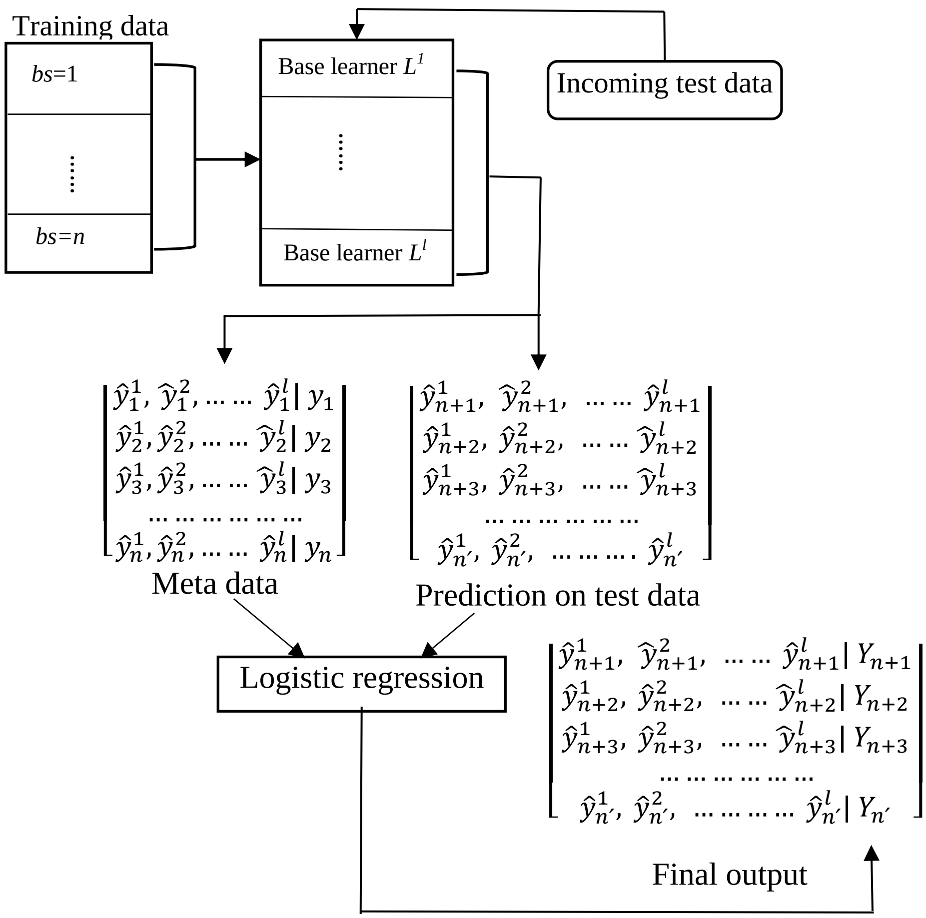

Figure 1: Prediction methodology of the Heterogeneous Stack Ensemble (HSE) model

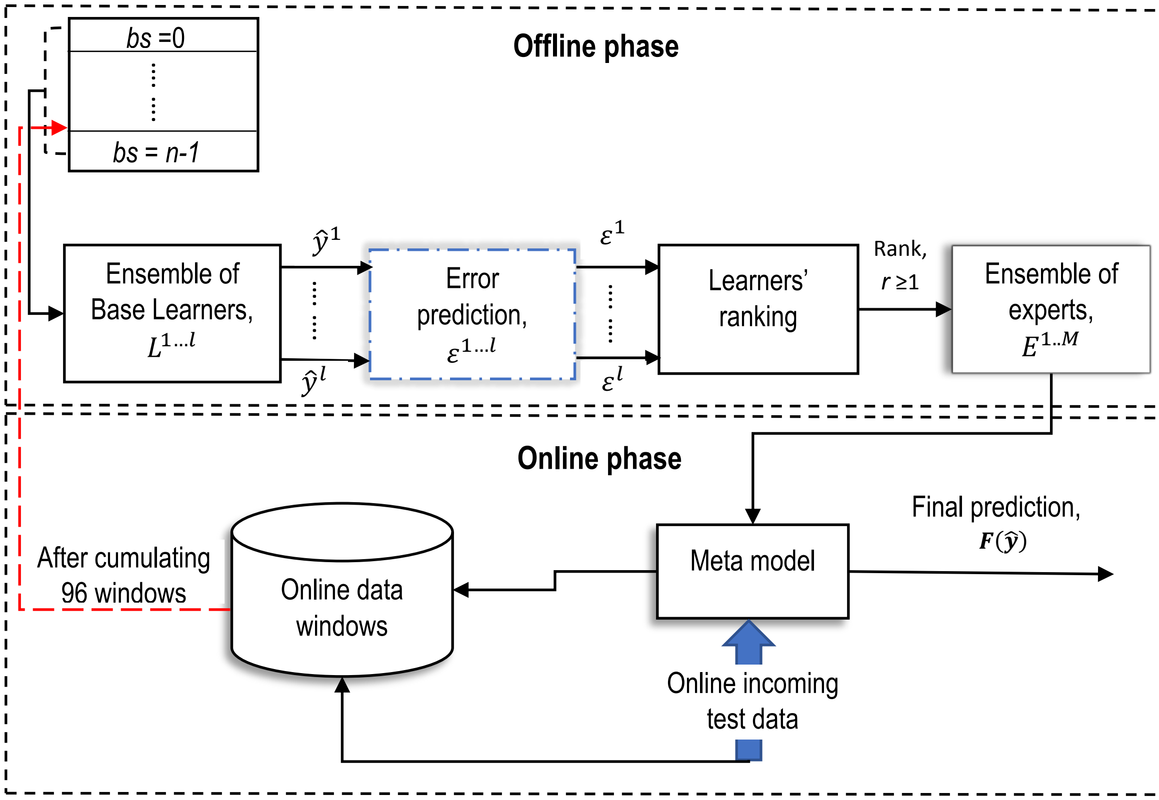

Figure 2: Block diagram of the Adaptive Batch-Ranked Ensemble (ABRE) model

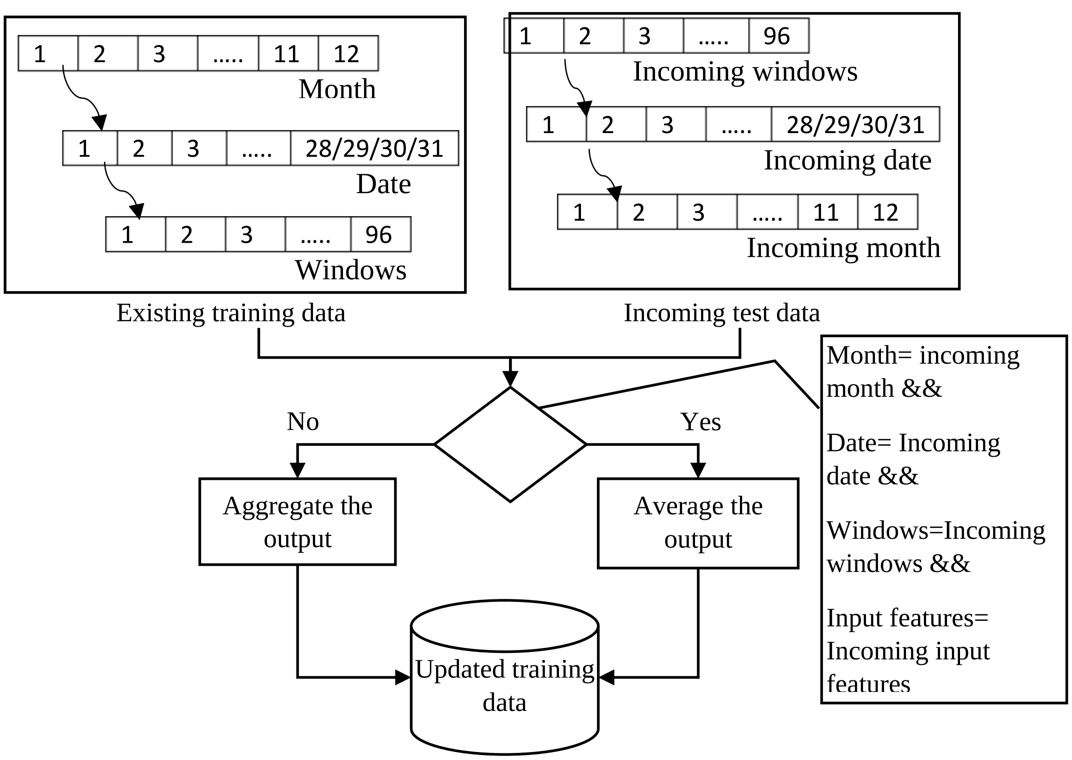

Figure 3: Aggregation and averaging method for time-variant training windows

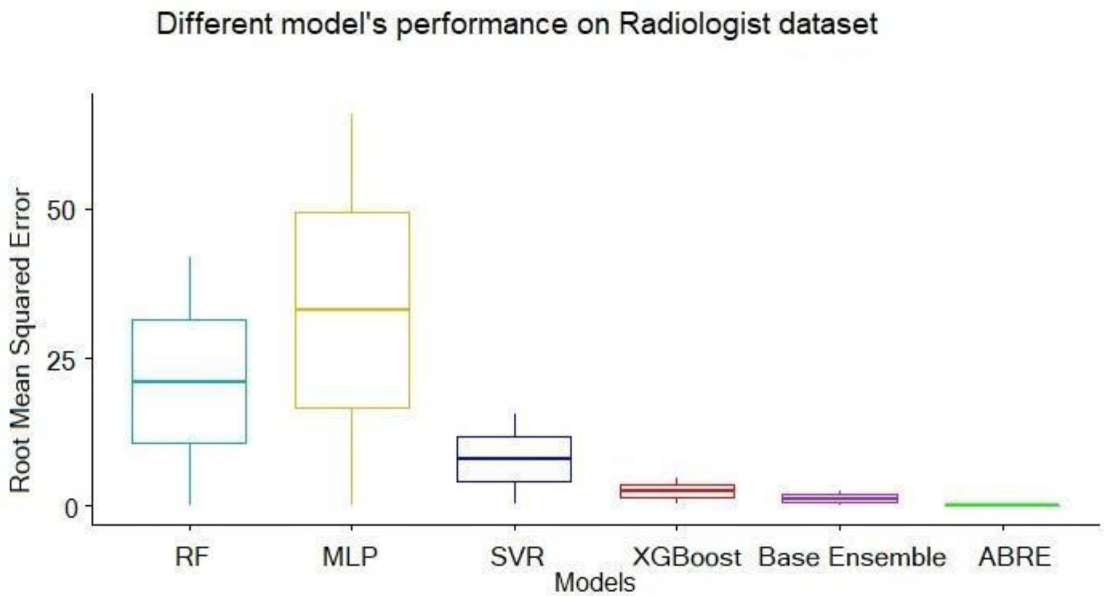

Figure 4: Different models' performance measure based on the RMSE

Tables:

Table 1: Review of related papers

Authors	Streaming data	Multivariate	Regression	Concept drift	Fluctuation reduction	Incremental learning
Sun, J., et al. (2017)	Y	Y	N	Y	N	N
Zhu, L., et al. (2018)	N	N	N	N	Y	Y
Qiu, X., et al. (2018)	Y	N	Y	Y	N	Y
Yu and Webb (2019)	N	Y	N	Y	N	Y
Ren, S. et al. (2019)	Y	Y	N	Y	N	Y
Xiao, J., et al. (2019)	Y	N	Y	Y	N	Y
Idrees, M. M. et al. (2020)	Y	Y	N	Y	N	Y
Li, Z. et al. (2020)	Y	Y	N	Y	N	Y
Ancy, S., & Paulraj, D. (2020)	Y	N	N	Y	Y	Y
Wang, J., et al. (2020)	Y	N	Y	N	Y	Y
Ahmad et al. (2021)	N	N	Y	N	N	N
Priya and Uthra (2021)	Y	N	N	Y	N	N
Yang et al. (2021)	Y	N	N	Y	Y	Y
Chen et al. (2021)	Y	Y	N	Y	N	N
Micevska et al. (2021)	Y	Y	N	Y	N	N
Mizan & Taghipour, S (2021)	Y	Y	Y	N	N	Y
Proposed ABRE model	Y	Y	Y	Y	Y	Y

Table 2: List of rank-based expert learners

Base learners(L^l)	Attained rank 1(r_j)	Expert learners(E^M)	Weight(w_j)
RF (number of tree 50)			
RF (Number of tree 100)	5,028	Random forest (#tree 100)	0.1431
RF (Number of tree 150)			
ANN (#hidden nodes 5)			
ANN (#hidden nodes 7)			
ANN (#hidden nodes 10)			
SVR (Linear kernel)			
SVR (RBF kernel)	19651	SVR (RBF kernel)	0.5593
SVR (Polynomial kernel)			
SVR (Laplace kernel)			
GBM (depth 100)			
XGBoost (depth 100)	9862	XGBoost (depth 100)	0.2806
MLP (#hidden layers and nodes 2[7,5])			
MLP(#hidden layers and nodes 3[10,7,5])	595	MLP (3[10,7,5])	0.0169
GLM (0.5,1)			

Table 3: Performance comparison between stand-alone models and ensemble models

Predictive Model	Execution time (millisecond)	Average RMSE for the next 96 test frames	Average MASE for the next 96 test frames
Random Forest	273.2	23.89	21.9
Support Vector Regressor	259.5	19.34	10.1
Extreme Gradient Boosting	298.1	11.3	6.7
Multilayer Perceptron	339.5	33.5	16.4
HSE model	394.8	8.73	6.5
Proposed ABRE Model	268.7	2.98	1.57

Table 4: Performance comparison of the ABRE model on some public datasets

Datasets	Instances	Features	HSE model's Execution time (ms)	HSE Model's RMSE	ABRE Execution time (ms)	ABRE Model's RMSE	% of Improvement in runtime	% of the Decrease in RMSE
Canadian_Climate_Normals.csv	30984	16	33.05	4.27	23.09	2.39	30.12	44.14
AirQualityUCI	9358	15	9.36	0.97	6.84	0.65	26.91	33.06
Appliances energy prediction Data Set	19735	29	38.15	3.29	27.73	3.22	27.31	1.94
Power System Data_60 min	125594	393	3290.56	243.14	1702.01	16.93	48.27	93.04
Behavior of the urban traffic of the city of Sao Paulo in Brazil	135	18	0.16	0.01	0.13	0.051	21.60	- 412
Open Power System_30 min	251187	56	937.76	44.09	425.05	4.79	54.67	89.13
Bike Sharing Dataset-Hourly	17389	16	18.55	1.07	12.59	0.72	32.10	31.98
Parking Birmingham	35717	4	9.52	0.61	6.93	0.46	27.28	24.71
Open Power System Data_15 min	502373	19	636.34	47.02	269.14	4.22	57.71	91.03
ICU	7931	14	7.40	0.64	5.83	0.33	21.26	48.12

Author biography

Tasquia Mizan Tasquia Mizan is a PhD candidate at the Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, Ontario, Canada. She is also associated with Reliability, Risk and Maintenance Research Laboratory of the same university. Before that, Tasquia Mizan worked as a lecturer in the Department of Computer Science and Information System, Najran University, Najran, KSA.

Her research interests comprise of data mining, machine learning, predictive forecasting, text classification and text mining, sentiment analysis, image analysis, and different branches of analytical methods. She has received Ryerson Graduate Scholarship in 2019 and some research awards in the past few years.

Sharareh Taghipour Sharareh Taghipour is an Associate Professor at the Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, Ontario, Canada and The Director of Reliability, Risk and Maintenance Research Laboratory of the same university. She is Canada Research Chair in Physical Asset Management. She obtained her PhD in Industrial Engineering from the University of Toronto and received her BSc in Mathematics and Computer Science and her MASc in Industrial Engineering, both from Sharif University of Technology, Iran.

Dr. Taghipour has received numerous awards, including Ontario Ministry of Research and Innovation –Early Researcher Award (2019), Ryerson University Early Research Career Excellence Award (2017), American Society for Quality (ASQ)-Reliability Division (RD) Best Paper Award (2016), Ryerson Faculty of Engineering and Architectural Science Scholarly Research and Creative Activity (SRC) Award (2015), The Best Student Paper Award of the Tom Fagan Reliability & Maintainability Symposium (2011), The Best Student Paper Award of the American College of Clinical Engineering (2010), and Asset Management Council Postgraduate Research Award (2010) from Australia.