Towards Online Anomaly Detection in Steel Manufacturing Process

Jakub Jakubowski[1,2][0000-0002-4773-9086], Przemysław Stanisz[2], Szymon Bobek[3][0000-0002-6350-8405], and Grzegorz J. Nalepa[3][0000-0002-8182-4225]

[1] Department of Applied Computer Science, AGH University of Science and Technology, 30-059 Krakow, Poland
[2] ArcelorMittal Poland, 31-752 Krakow, Poland
[3] Jagiellonian University, Faculty of Physics, Astronomy and Applied Computer Science, Institute of Applied Computer Science, and Jagiellonian Human-Centered AI Lab (JAHCAI), and Mark Kac Center for Complex Systems Research, ul. prof. Stanisława Łojasiewicza 11, 30-348 Kraków, Poland

Abstract. Data generated by manufacturing processes can often be represented as a data stream. The main characteristics of these data are that it is not possible to store all the data in memory, the data are generated continuously at high speeds, and it may evolve over time. These characteristics of the data make it impossible to use ordinary machine learning techniques. Specially crafted methods are necessary to deal with these problems, which are capable of assimilation of new data and dynamic adjustment of the model. In this work, we consider a cold rolling mill, which is one of the steps in steel strip manufacturing, and apply data stream methods to predict distribution of rolling forces based on the input process parameters. The model is then used for the purpose of anomaly detection during online production. Three different machine learning scenarios are tested to determine an optimal solution that fits the characteristics of cold rolling. The results have shown that for our use case the performance of the model trained offline deteriorates over time, and additional learning is required after deployment. The best performance was achieved when the batch learning model was re-trained using a data buffer upon concept drift detection. We plan to use the results of this investigation as a starting point for future research, which will involve more advanced learning methods and a broader scope in relation to the cold rolling process.

Keywords: data streams · anomaly detection · cold rolling.

1 Introduction

Progressing digitalization of the industry has led to the production of enormous amounts of data, which possesses many characteristics of data streams. These data are generated in form of a continuous flow of information from sensor readings, which is produced with high speed and volume and is infinite in nature, as new data will be generated as long as the manufacturing process is in operation.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

In many cases, it is not possible to store all sensor readings and process them in batch mode. This approach might be infeasible due to hardware or software constraints. The manufacturing process may evolve over time (concept drift) due to factors like wear of the asset, changes in production mix, or modifications in the production process. Still, sensor data may give valuable insight into the nature of the process and improve its performance by e.g., adapting the process to new conditions or detecting anomalies in near real time. Therefore, it can be very beneficial for companies to use these data; however, its processing pipeline should be well adjusted to the specific problem and take into account important characteristics of the data.

Learning from data streams is a challenging task that involves dealing with many data problems, i.e., inability to process all data at once, variations in data distribution over time, class imbalance, delayed or inaccessible labels [13]. It often requires the assimilation of most recent data to adapt the model to dynamically changing conditions. To build a robust machine learning model to control the manufacturing process, all the mentioned factors must be carefully analyzed and addressed.

In this work, we present our preliminary results on the machine learning model that controls the steel cold rolling process and detects potential anomalous measurements. Cold rolling is an important step in the steel manufacturing process, where the thickness of the steel is reduced to reach the dimensions requested by the client. One of the critical parameters of this process is the rolling force, which should be carefully monitored to ensure the quality of the final product [24]. To control the rolling forces, we propose a machine learning approach, which learns the proper distribution of forces based on the given steel properties and rolling parameters. The main goal of the proposed solution is to detect anomalies in the rolling force in real time. We exploit three different scenarios of the ML pipeline to handle the problem and evaluate them in terms of their learning capabilities. Our baseline scenario assumes processing the small part of the data in batch mode and evaluating its results on the rest of the data. In the second scenario, we retrain the model in batch mode using a sliding window every time a drift is detected. Finally, we use online ML models to learn from the data continuously as each data point arrives. We also discuss and tackle the problem of data imbalance, which is present in our dataset and is one of the important concerns when learning from streaming data. To the best of our knowledge, there is no paper dealing with the problem of anomaly detection in cold rolling by using data stream learning techniques. One of the issues that we discuss in this paper is the problem of distinction between anomalies and concept drift that might occur in the data stream.

The rest of the paper is organized as follows. In Section 2 we briefly discuss state-of-the-art techniques for learning from data streams and provide details on the cold rolling mill. In Section 3 we present the details of the proposed anomaly detection model and the learning scenarios. In Section 4 we present our preliminary results from the experiments carried out and discuss them. In

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

Section 5 we conclude our findings and discuss the potential directions of future research in this topic.

2 Related Works

2.1 Data Streams

Learning from data streams is a well-established problem in the field of machine learning and has been a topic of many studies [10, 11]. The main requirements for an ML model in streaming data setting that need to be faced include: (1) the computation time and memory usage should be irrespective of the total number of observed measurements, (2) the model should be able to learn by using each observation only once, (3) the model should perform similarly to equivalent model trained in batch setting, and (4) the model should be able to adapt to any changes in the data distribution, but should not forget relevant information from the past (catastrophic forgetting). In practice, many of these requirements are difficult to meet as some trade-offs might occur. For example, increasing the complexity of the model may allow to increase its accuracy but may deteriorate its ability to learn in real-time. Models such as neural networks benefit greatly from making many passes (epochs) over the same chunk of data, so limiting learning to a single pass over data will usually decrease their performance.

Many state-of-the-art machine learning models, e.g., XGBoost, Random Forest, require access to the entire dataset for training. After the training has ended, it is not possible to adjust the model to the new data while preserving some part of the information learned earlier. To solve this problem, specific algorithms for online learning, which can learn incrementally as new data arrive, were proposed [12, 15]. On the other hand, ordinary ML algorithms can also be useful in streaming data applications if we are able to store some part of data in memory in the form of, e.g. sliding window.

Concept dirft is one of the biggest problems faced when dealing with streaming data. Changes in data can be *actual*, when there occurs a change in the decision boundaries, or *virtual*, when the decision boundary remains stable, but the distribution of the data within each class changes [21]. Many concept drift detection methods have been proposed, such as EDDM [2], ADWIN [3] to list a few. They focus on observing the performance of the model and detecting the point in time in which the model starts to deteriorate. This is an indication that the current model no longer preserves its original accuracy and that some adaptations are needed to recover the model.

The problem of data imbalance is well known and has been studied, especially in batch learning [14]. However, in streaming data, it is particularly difficult to deal with class imbalance due to factors like dynamically changing distribution of the data. The class imbalance may also affect the drift detection algorithms [6] where, for example, the classifier has a lower accuracy on rare examples. There are several strategies to deal with imbalanced data. Oversampling strategies aim to generate new samples from minority classes, while undersampling selects only

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

some portion of the data from majority classes to achieve better data balance. There exist also hybrid approaches, which combine the above mentioned strategies.

2.2 Anomaly Detection

Anomaly detection methods aim to discover the outliers in a dataset, that do not fit to the observed distribution and usually constitute a small fraction of the data. In supervised learning scenarios, where each observation has an anomaly label assigned, the problem can be described as a specific type of imbalanced learning with binary classification task. However, in practical applications, especially in manufacturing processes, the data are usually generated without labels. In such cases, it is not possible to use supervised learning techniques.

On the other hand, unsupervised learning methods are very robust in anomaly detection tasks. There exist several well-studied techniques for anomaly detection based on tree or clustering algorithms, which also have their implementation for online learning problems [25, 26]. Another approach to detect anomalies is to train a regular ML model to learn the relationship between independent and dependent variables and assume the error of the model as the anomaly level.

2.3 Cold Rolling Process

ArcelorMittal Poland is the largest steel producer in Poland, and its production chain includes all relevant steps from the production of pig iron to the final product in the form of a steel strip. In the steel plant, the thickness of the steel is about 220 mm. One of the important steps in the production of steel strips is to reduce this thickness to obtain the product ordered by customers. First, the steel is hot rolled, where the thickness of the steel is reduced at high temperature to a range of about 2 to 5 milimeters. If there is a demand for a lower thickness, a cold rolling is necessary. During cold rolling, a metal strip is passed through a rolling mill, consisting of many subsequent pairs of work rolls without prior preheating. The minimum strip thickness produced in the analyzed plant is below 0.4 mm. The simplified scheme of the considered rolling mill is presented in Figure 1.

During contact of the strip with the work rolls, a force is applied to the rolls, which is transferred to the strip and causes its reduction in thickness and equivalent elongation. In steel manufacturing mills, the cold rolling is performed at very high speeds (sometimes exceeding 1000 m/min), which requires a very fast adaptation of process parameters to achieve the desired thickness and flatness of the final product. These adaptations are usually made by a PLC, that controls the whole process. The superior goal of the control system is to achieve desired thickness after the last stand, while preserving the safe limits of the speed, forces, etc.

There are several process parameters that can be used to evaluate the condition of the rolling mill. One of the most important parameters is the rolling force applied in each stand. In general, these forces should be within certain limits based on the product characteristics (thickness, width, steelgrade) and

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

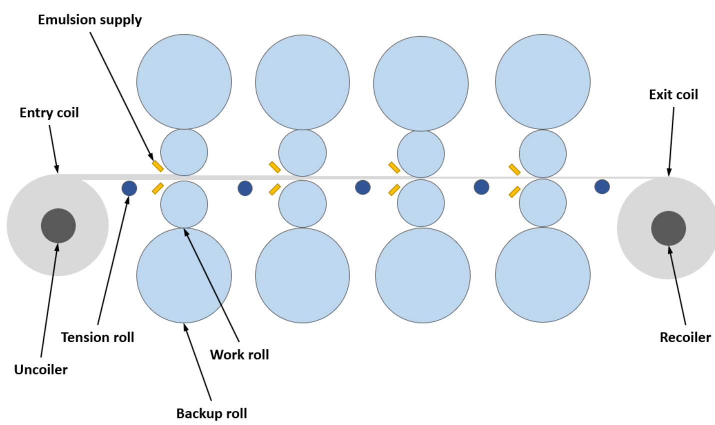

Fig. 1. Schematic diagram of tandem cold mill with four rolling stands.

process parameters (speed, state of the work rolls). High deviations of the forces from the normal working condition may be a symptom of malfunctions and can lead to dangerous situations, e.g., strip breaks. However, taking into account the number of variables that influence the force value, it is infeasible to build simple rules to control their values in a kind of manual mode.

Rolling mechanics has been studied for decades, and there are several state-of-the-art physical models of this process [1, 4]. The main problem with the physical models is that they usually rely on some assumptions, which are hard to precisely determine during online production. For example, the Bland and Ford model [4] is highly dependent on friction between the strip and the work roll, which is a kind of stochastic parameter and is almost unmeasurable in the production environment [20]. This implies that more robust machine learning approaches could be used to model the cold rolling process. Several researchers proposed using ML methods in the analysis of the cold rolling process [8, 16, 19], but we have not found work that used the approach of learning from streaming data. The aforementioned characteristics of the data streams are consistent with the characteristics of the steel rolling process. Imporant process parameters, for example, steel yield strength, friction coefficient, are prone to systematic variations due to changes in roll roughness, lubrication, or unknown variations in previous production steps. Therefore, it is important that the model developed for the cold rolling process is robust and has the ability to evolve under changing conditions. On the other hand, the model should remember relevant patterns from the past – this is important especially for the rarely produced steels, which can appear in the schedule only a few times in a year.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

3 Research Methods

3.1 Dataset description

We have collected historical data from the Cold Rolling Mill in Kraków, Poland. The data set contains records of 20,000 steel coils, representing a few months of production. The original data are generated by a process with a sampling frequency of 1Hz. The data were filtered from the periods when the production line was stopped and when impossible values (from a process point of view) were recorded. However, we tried to keep anomalous measurements within the data, as the goal of the proposed solution is to detect outliers. We have manually scaled the data, based on known process limits, to fit the observations within the range 0-1. The objective of the ML model is to accurately predict the required rolling forces in the first three stands given the material properties, rolling speed, reduction, and tensions. We remove all the variables which are correlated with force but are not independent of it, e.g., rolling gap, motor torque, and motor load. The final dataset consists of over 250,000 observations and 23 features. Next, on the basis of the mean squared error (MSE) between the predicted and measured rolling forces, the anomaly score is calculated.

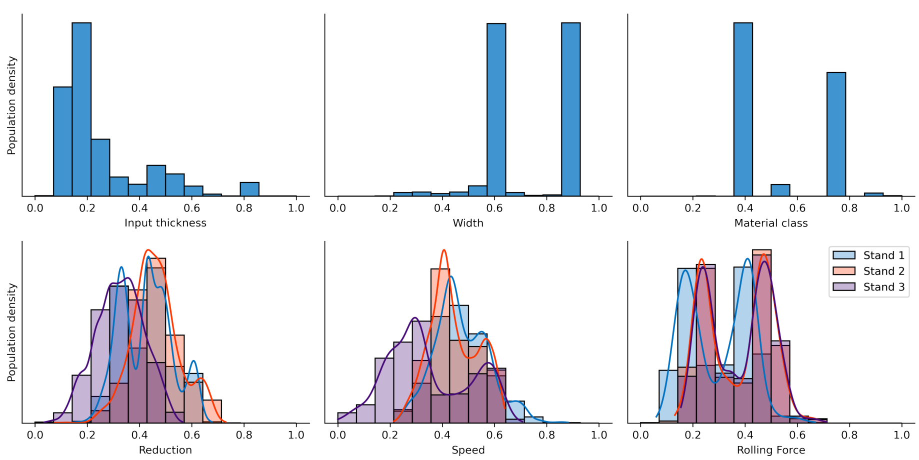

Fig. 2. Normalized data distribution of the most relevant features

Figure 2 presents the distribution of selected features. These distributions show that we have a potential issue with inbalanced data, since most of the production is a thin material with a very specific width and material class. We address this issue and describe our solution in Section 3.2.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

3.2 Dealing with imbalanced data

In Section 3.1 we discuss the problem of data imbalance, which is caused by the fact that the majority of production consists of few types of products and the other products are not well represented in the dataset. This could easily lead to overfitting the model to a majority group of products.

To solve this problem, we propose to use a clustering algorithm to group the data based on intrinsic product characteristics, that is, thickness, width, and material class. We use the KMeans model, which is built on the training dataset to assign each observation to a cluster. The number of clusters is determined based on the Silhouette Score [23].

In the next step, we have used a sampling algorithm, which combines the under-sampling and over-sampling methods. We have iterated over each point and computed the repetitions of that sampled point on the basis of the cluster size and the Poisson distribution. In practice, some of the samples from overrepresented clusters were omitted, and the samples from underrepresented clusters were duplicated. To avoid generating too many of the very same points, we limited the number of possible sample repetitions to three.

3.3 Learning scenarios

To simulate the data stream environment, we have divided the dataset into the following proportions: 15% for training, 5% for validation (hyperparameter tuning) and 80% for testing. It is important to mention that although we divide the data in an ordinary way, in some scenarios the test data is used to learn the model, e.g. in online learning where the model never stops learning. In this work, we analyze three scenarios regarding the model learning process in order to find the best solution to the problem.

Batch model This is the baseline scenario, which assumes that we follow an ordinary machine learning pipeline, where the model is trained only once on a training dataset. This scenario assumes that we are able to collect enough data during the model development and that no concept drift will occur in the future.

Batch model with retraining In this scenario, we train the model as described in the above scenario. The trained model is put into production (run on a test dataset) along with a concept drift detection algorithm, which controls if the current model is still valid. We also keep a fixed-size buffer in the form of a sliding window (FIFO). If at any point a drift is detected, the current model is dropped and a new model is built, which is trained on the data stored in the buffer. Therefore, the assimilation of the new data is made upon detection of concept drift.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

Online learning In this setting, we apply an online learning paradigm, where the model is learned iteratively as new samples arrive. This scenario does not require separation of the data into training, validation, and test sets, but we keep such a structure to be more coherent with previous scenarios to allow for a trustworthy evaluation of the results. In this setting, the assimilation of the recent data is made as soon as it is accessible. Since we assume that we have access to the data stored in the training and validation sets, we can calculate the performance metrics on the validation set to perform hyperparameter tuning.

3.4 Learning algorithms and validation

In this work we have used three different types of machine learning algorithms, that is:

- Linear Model (with L1 regularization)
- Multi-Layer Perceptron (MLP), which is one of the basic types of Artificial Neural Networks (ANN)
- Random Forest (RF).

In the latter case, more specifically, we have used a forest of CART [5] in a batch setting and Adaptive Random Forest [12] (ARF) built with Hoeffding Trees [9] in an online setting. We have utilized *sckit-learn* [7] library for training batch models and *river* [22] library for online learning. The choice of models was mainly motivated by the fact that all three types of models have their implementation for batch and online learning. The focus of this work is on selecting the best learning scenario, rather than achieving the best possible performance of the model.

The hyperparameters of each model were determined using a grid search method by fitting the model to the training dataset and evaluating its mean absolute error (MAE) in the validation data set. We have chosen MAE as the main metric for model evaluation to minimize the effect of anomalies, which we want to find in the next step. However, we also evaluate the models based on the root mean squared error ($RMSE$) and the coefficient of determination (R^2).

An important aspect of model evaluation is the proper selection of observations for the calculation of performance metrics. In our use-case we deal with situations where we observe very similar points for a longer period of time due to e.g. production of the same steel strip for a longer period of time. Such a setting results in situation where dummy predictor, which uses previous measurement as the prediction, may have very high performance metrics. To address this issue, we have limited the data used for evaluation purposes. The condition to include the observation in the evaluation data was that there is a significant difference between the current and previous force measurements (set at 3% of the total range).

3.5 Anomaly detection

The anomaly detection is performed by comparing the measured rolling forces with the values predicted by the model. This assumes that the model has learned

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

the relationship between input and output features precisely. The difference between those values can be treated as the anomaly level. The magnitude of the anomaly, which we will refer to as the *anomaly score*, is computed by calculating MSE between the observed and predicted forces.

An important aspect, which we consider in our work, is that we must be able to distinguish between anomalies and concept drifts. In some cases the rapid changes in the value of the force may not be a clear symptom of anomaly, but a sudden drift which occurred in the process. We note that sometimes it might not be possible to distinguish between them, as a concept drift might be a result of some persistent failure (which from the process point of view should be treated as an anomaly). Moreover, we assume that the model should not update itself on anomalous samples; therefore, once detected, they should not be included in the model learning in the online phase. To resolve these issues, we propose the following methodology. First, we determine the running statistics of the mean anomaly score (μ) and its standard deviation (σ). Next, we compare the anomaly score of a signle observation with the distribution of all calculated anomaly scores.

1. If the anomaly score is less than $\mu+2\sigma$, there is no anomaly, and any ongoing deviations can be treated as concept drift.
2. If the anomaly score is between $\mu + 2\sigma$ and $\mu + 3\sigma$, there is a high risk that we observe an anomaly, but we consider this as a transient state and only raise a warning. The model is not updated with this data.
3. If the anomaly score is above $\mu + 3\sigma$, we raise an alert, which means that the anomaly is detected.

Nevertheless, we use the observations marked as anomalies in the calculation of model performance metrics to avoid the situation when the incorrectly learned relations are discarded from the validation.

4 Results and Discussion

In this section, we present the results of our research. The first step of the machine learning pipeline was the resampling of the original data. To determine which observations to undersample and which to oversample, we have clustered the observations, as discussed in 3.2, with the KMeans algorithm. For our application, the highest Silhouette score was achieved when the number of clusters was $n = 6$. Figure 3 presents the number of observations in the training set before and after resampling.

Below, we present the achieved performance metrics of each model on the test dataset – each model was verified in terms of its mean absolute error, root mean squared error, and coefficient of determination. We also evaluated each model in terms of the share of anomalies, which is the number of anomalies (determined as described in Section 3.5) as a fraction of all observations. The results are listed in Table 1.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

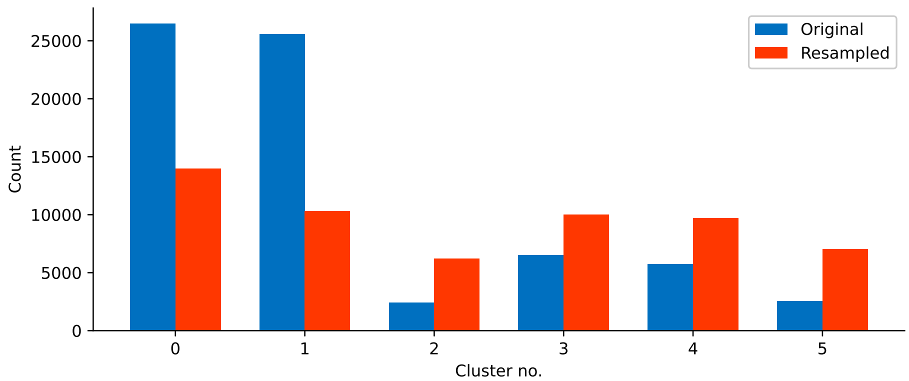

Fig. 3. Count of original and resampled data with respect to the assigned cluster.

The results show that the best performance of the model has been achieved in a setting with a batch model and a concept drift detector. Suprisingly, the best metrics were achieved for Linear Models, which are characterized by the least complexity. This induces the correlations between the variables to be mostly linear, and increasing the complexity of the model might harm the ability of the model to generalize. However, the metrics achieved for the MLP network were comparable, indicating that some more fine-tuning of this algorithm could help outperform the Linear Model. The most important observation is that there is a significant increase in performance of the models, if the model learns continuously. This implies that a model without the ability to adapt as new data come is not able to precisely predict the rolling force and thus potential anomalies. Such a model in production could result in too many false alarms, which is an undesirable scenario.

The proposed methodology for anomaly detection results in having between 1.0% and 2.5% of alarms. These values seem to be in acceptable range; however, this problem should be further addressed to determine optimal anomaly threshold for the analyzed process. We observe that online models tend to give less alarms than other methods, which is most probably due to its ability to quickly adapt to changing conditions.

Figure 4 presents how the anomaly score varies with the number of observations. In the initial stage, we note that all three models have a similar anomaly score. However, after approximately 12,000 observations, a sudden drift occurs, leading to an increase in the anomaly score for batch models. The online model maintains a similar anomaly score as before. Subsequently, the batch model equipped with drift detection is capable of adapting to new data and reducing the anomaly score, which is the situation we expected.

Figure 5 presents the sample of data, where the rolling force in the first stand is plotted over time. Comparison of the measured force with the calculated ones clearly shows that at a given point in time the accuracy of the unadapted

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

Table 1. Achieved model performance metrics and corresponding percentage of observations marked as anomalies. The best results are highlighted in bold.

Method	**ML model**	**Metrics**			**Anomaly share**
		MAE	RMSE	R^2	
Batch Learning	Linear	0.052	0.066	0.74	1.6%
	ANN	0.050	0.063	0.76	2.5%
	RF	0.047	0.059	0.79	2.2%
Batch Learning with Retraining	Linear	**0.033**	**0.044**	**0.89**	1.8%
	ANN	0.035	0.046	0.88	1.9%
	RF	0.038	0.051	0.85	2.2%
Online Learning	Linear	0.035	0.046	0.88	1.4%
	ANN	0.038	0.051	0.85	1.4%
	ARF	0.036	0.047	0.87	1.0%

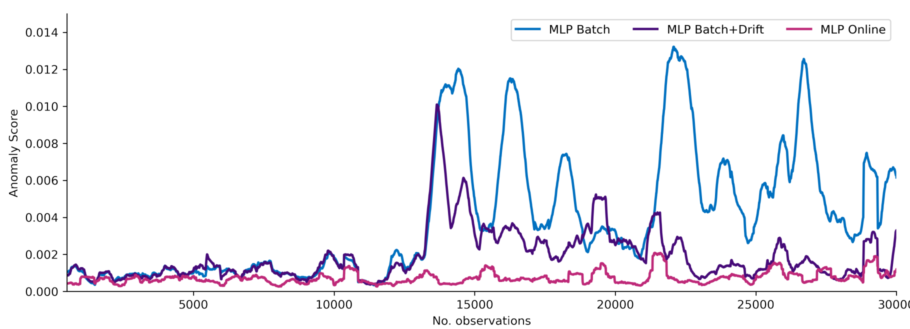

Fig. 4. Examplary anomaly score variations (moving average) for the MLP models.

batch model is much worse than that of the other two models. Furthermore, a discovered anomaly region has been highlighted, where the force has largely increased for a certain period of time, but the models did not predict it to happen, showing that they are able to indicate the anomaly in the process.

5 Conclusion and Future Works

In this paper, we have applied machine learning models to predict rolling forces in a cold rolling process. The goal of this model is to control the manufacturing process online and detect anomalies. The characteristics of the problem imply the use of methods designed for data streams. The main data stream features, which we considered were the inability to store a whole data set in memory, the possibility of concept drift occurring, and imbalances in the data. We have considered three learning scenarios: (1) batch learning without any changes in

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

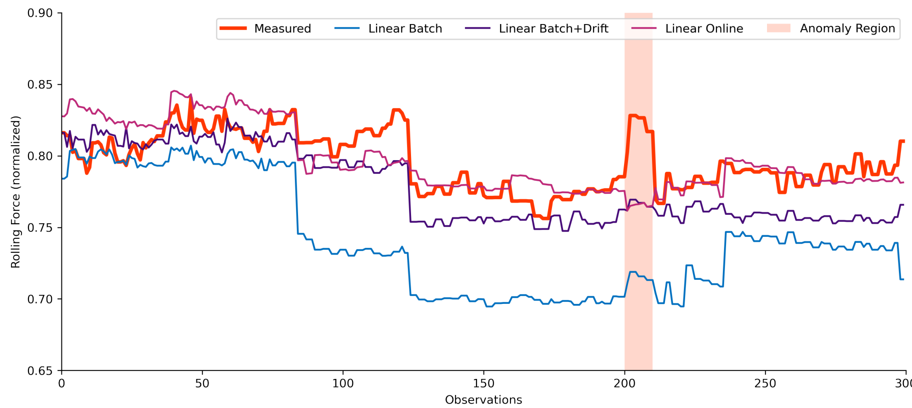

Fig. 5. Examplary rolling forces in the first stand measured and calculated by the linear models.

the model once it is deployed in production, (2) batch learning with concept drift detection and retraining upon the changes in the data distribution, and (3) online learning, where each sample is seen only once by the model. Three different types of machine learning models were explored; linear model, artificial neural network, and random forest.

The results have shown that the baseline scenario, which did not take into account the possibility of concept drift, has shown poor performance in comparison with the other scenarios. The best results were obtained in the scenario where the batch model was re-trained on a buffer data every time a drift has been detected. However, the online learning scenario has also shown some promising results, so this option should also be considered as a potential solution to our problem. Our work shows that machine learning from streaming data has a high potential to discover anomalies in the cold rolling process. Assimilation of recent data is crucial to keep high prediction capabilities of the model. Although our work is strictly dedicated to the steel manufacturing process, we believe that a similar approach can be adapted to other manufacturing processes.

In future work, we plan to further investigate the applications of data stream methods to a problem of anomaly detection in steel manufacturing processes. We plan to focus on the problems that were discovered in this investigation but can be analyzed more in-depthly. First, we want to investigate the use of ensemble learners, which are one of the most promising methods for dealing with drifting data streams [18] and are capable of solving issues such as pattern forgetting or overfitting. Next, we plan to investigate more robust methods for dealing with class imbalance and buffer storage, which will result in having more diverse training data and better generalization. When it comes to online learning, we want to validate whether the methods designed to tackle the problem of catastrophic forgetting, for example, EWC [17], are able to help online algorithms outperform their batch equivalents. Finally, we want to extend our method by

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37

detecting anomalies not only in the rolling force measurements but also in other dependent variables such as gap position or motor load.

Acknowledgements Project XPM is supported by the National Science Centre, Poland (2020/02/Y/ST6/00070), under CHIST-ERA IV programme, which has received funding from the EU Horizon 2020 Research and Innovation Programme, under Grant Agreement no 857925.

References

1. Alexander, J.M.: On the theory of rolling. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences **326**(1567), 535–563 (1972), http://www.jstor.org/stable/77929
2. Baena-García, M., Campo-Avila, J.D., Fidalgo, R., Bifet, A., Gavalda, R., Morales-Bueno, R.: Early drift detection method (2005)
3. Bifet, A., Gavaldà, R.: Learning from time-changing data with adaptive windowing (2007)
4. Bland, D.R., Ford, H.: The calculation of roll force and torque in cold strip rolling with tensions. Proceedings of the Institution of Mechanical Engineers **159**(1) (Jun 1948)
5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees (1984)
6. Brzezinski, D., Stefanowski, J., Susmaga, R., Szczech, I.: On the dynamics of classification measures for imbalanced and streaming data. IEEE Transactions on Neural Networks and Learning Systems **31**, 2868–2878 (8 2020). https://doi.org/10.1109/TNNLS.2019.2899061
7. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013)
8. Chen, Z., Liu, Y., Valera-Medina, A., Robinson, F.: Strip snap analytics in cold rolling process using machine learning. In: 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE). pp. 368–373 (2019). https://doi.org/10.1109/COASE.2019.8842967
9. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 71–80. KDD '00, Association for Computing Machinery, New York, NY, USA (2000). https://doi.org/10.1145/347090.347107
10. Domingos, P., Hulten, G.: Catching up with the data: Research issues in mining data streams (2001)
11. Gama, J.: Knowledge Discovery from Data Streams. 1st edn. (2010)
12. Gomes, H.M., Bifet, A., Read, J., Barddal, J.P., Enembreck, F., Pfharinger, B., Holmes, G., Abdessalem, T.: Adaptive random forests for evolving data stream classification. Machine Learning **106**, 1469–1495 (10 2017). https://doi.org/10.1007/s10994-017-5642-8
13. Gomes, H.M., Read, J., Bifet, A., Barddal, J.P., Gama, J.: Machine learning for streaming data. ACM SIGKDD Explorations Newsletter **21**, 6–22 (11 2019). https://doi.org/10.1145/3373464.3373470

14. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications **73**, 220–239 (5 2017). https://doi.org/10.1016/J.ESWA.2016.12.035
15. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 97–106. KDD '01, Association for Computing Machinery, New York, NY, USA (2001). https://doi.org/10.1145/502512.502529
16. Jakubowski, J., Stanisz, P., Bobek, S., Nalepa, G.J.: Roll wear prediction in strip cold rolling with physics-informed autoencoder and counterfactual explanations. Proceedings of The 9th IEEE International Conference on Data Science and Advanced Analytics (DSAA) (2022)
17. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017). https://doi.org/10.1073/pnas.1611835114, https://www.pnas.org/doi/abs/10.1073/pnas.1611835114
18. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: A survey. Information Fusion **37**, 132–156 (9 2017). https://doi.org/10.1016/j.inffus.2017.02.004
19. Lee, S., Son, Y.: Motor load balancing with roll force prediction for a cold-rolling setup with neural networks. Mathematics **9**(12), 1367 (Jun 2021). https://doi.org/10.3390/math9121367
20. Lenard, J.G.: 9 - tribology. In: Lenard, J.G. (ed.) Primer on Flat Rolling (Second Edition), pp. 193–266. Elsevier, Oxford, second edition edn. (2014). https://doi.org/https://doi.org/10.1016/B978-0-08-099418-5.00009-3
21. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., Zhang, G.: Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering pp. 1–1 (2018). https://doi.org/10.1109/TKDE.2018.2876857
22. Montiel, J., Halford, M., alaneu Alan, mastelini, F.S.M.M., Bolmier, G., Vaysse, R., Zouitine, A., Gomes, H.M., Read, J., Bifet, A., Mastelini, S.M., Sourty, R., Abdessalem, T.: River: machine learning for streaming data in python (2021), https://github.com/online-ml/river.
23. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987). https://doi.org/https://doi.org/10.1016/0377-0427(87)90125-7, https://www.sciencedirect.com/science/article/pii/0377042787901257
24. Rusnák, J., Malega, P., Svetlík, J., Rudy, V., Šmajda, N.: The research of the rolling speed influence on the mechanism of strip breaks in the steel rolling process. Materials **13**(16), 3509 (Aug 2020). https://doi.org/10.3390/ma13163509
25. Tan, S.C., Ting, K.M., Liu, T.F.: Fast anomaly detection for streaming data. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two. p. 1511–1516. IJCAI'11, AAAI Press (2011)
26. Yin, C., Zhang, S., Yin, Z., Wang, J.: Anomaly detection model based on data stream clustering. Cluster Computing **22**, 1729–1738 (1 2019). https://doi.org/10.1007/s10586-017-1066-2

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_37