



A Machine Learning Approach to Predict Chattering Alarms

Nicola Tamascelli^{a,b,*}, Tufan Arslan^c, Sirish L. Shah^d, Nicola Paltrinieri^b, Valerio Cozzani^a

^a Department of Civil, Chemical, Environmental and Materials Engineering, University of Bologna, Bologna, Italy

^b Department of Mechanical and Industrial Engineering, NTNU, Trondheim, Norway

^c Scientific Computing group, IT Department, NTNU, Trondheim, Norway.

^d Department of Chemical and Materials Engineering, University of Alberta, Alberta, Canada
 nicola.tamascelli@gmail.com

The alarm system plays a vital role to ensure safety and reliability in the process industry. Ideally, an alarm should inform the operator about critical conditions only and provide guidance to a set of corrective actions associated with each alarm. During alarm floods, the operator may be overwhelmed by several alarms in a short time span, and crucial alarms are more likely to be missed during these situations. Most of the alarms triggered during a flood episode are nuisance alarms –i.e. alarms that do not convey any new information to the operator, or alarms that do not require operator actions. Chattering alarms that repeat three or more times in a minute and redundant or duplicated alarms are common forms of nuisance alarms. Identifying such nuisance alarms is a key step to improve the performance of the alarm system. Recently, advanced techniques for alarm management have been developed to quantify alarm chatter; although effective, these techniques produce relatively static results. Machine learning algorithms offer an interesting opportunity to analyse historical alarm data and retrieve knowledge, which can be used to produce more flexible and dynamic models, as well as to predict alarms behaviour. The present study aims to develop a machine learning-based algorithm for chattering prediction during alarm floods. A modified approach based on run lengths distribution has been developed to evaluate the likelihood of future alarm chatter. The method has allowed categorizing historical alarm events as alarms that will (or will not) show chattering in the future. Finally, categorized alarms have been used to train a Deep Neural Network, whose performance has been evaluated against the ability to predict alarm chatter. Overall, the Neural Network has shown good prediction capabilities and most of the chattering alarms were correctly identified.

1. Introduction

The advent of the Distributed Control System (DCS) has undeniably improved flexibility and safety of chemical plants, but some issues have arisen as well. In the analog days, installing new alarms used to cost around 1000 \$/alarm (Katzel, 2007), including purchase and hard wiring of each alarm and the corresponding annunciator panels (Shaw, 1993). Nowadays, alarms are managed by the DCS. The cost for installing new alarms has dropped and physical panels are not required anymore (Katzel, 2007). The digitised installation has improved the flexibility of the alarm system but as a drawback, a large number of alarms are now present in most process system (Shaw, 1993). As a consequence, more than often the number of alarms displayed are unmanageable by the operator. Recently, standard manuals such as ANSI/ISA (2016) and EEMUA 191 (2013) have addressed the problem of poor alarm management in modern chemical plants, providing guidelines and suggestions. According to these standards, the average alarm annunciation rate should not exceed 6 alarms/hour per operator console to be considered manageable. Unfortunately, in most chemical plants, the alarms rate is much higher than the suggested value (Kondaveeti *et al.*, 2013).

Alarm floods are “conditions during which the alarm rate is greater than the operator can effectively manage (e.g. more than 10 alarms per 10 minutes)” (ANSI/ISA, 2016). During a flood episode, an operator may have to acknowledge and resolve hundreds of alarms in a short period. Clearly, an effective response is impossible in such a chaotic situation. Typically, a majority of the alarms in a flood episode are *nuisance* alarms (i.e. that

do not communicate any new information) (ANSI/ISA, 2016). Several types of nuisance alarms exist (e.g. chattering, fleeting and stale alarms). Chattering alarms are alarms “that repeatedly transitions between active state and inactive state in a short period of time” (ANSI/ISA, 2016). Therefore, chattering alarms have the potential to produce a large count of alarms and reducing their number is a key step to improve the performance of the alarm system during alarm floods. Kondaveeti *et al.* (2013) proposed a method for quantifying alarm chatter based on run lengths distributions. Although effective, this technique produces static results (i.e. chattering is quantified based on historical alarm data, but no conclusion can be drawn about the alarm future behaviour). In a modern context, where computer technologies and Industry 4.0 solutions are rapidly expanding among different sectors, the need for more dynamic and flexible models is real. In the current scenario, chemical plants produce and store an immense amount of data (Balasko and Abonyi, 2007), modern computers have outstanding calculation capability, and data science techniques have come a long way. We now have the technical capability and the tools to process a vast amount of data. However, process data is mainly archived and not analysed or explored to mine for information and knowledge. The availability of multivariate statistical and Machine Learning techniques now offers the opportunity to “learn” and extract knowledge from past data (Liu *et al.*, 2018).

For the reasons mentioned above, the objective of this study is to overcome the limitations of the existing methods for chattering quantification and to propose a Machine Learning based method for chattering prediction. Specifically, the Chattering Index approach proposed by (Kondaveeti *et al.*, 2013) has been modified to obtain a Dynamic Chattering Index, whose results are then used to train a Deep Neural Network model. The efficacy of the proposed method is evaluated by application to an industrial case data set consisting of alarm data from an ammonia production plant.

2. Alarms from ammonia production plant

An industrial alarm database has been considered to support the analyses. Specifically, alarm data from a section of an ammonia production process (Topsoe.com, 2020) is analysed. Due to the large quantity of hazardous substances stored and handled during normal activity, the plant has been classified as an “upper tier” Seveso III establishment. Extensive use of methane, hydrogen, and ammonia (anhydrous and aqueous solution) occurs in the plant section. Furthermore, due to the intrinsic properties of the processes involved, severe operating conditions (i.e. high pressure and high temperature) are often associated with corrosive substances. Additional information about ammonia production and the considered site can be found at: (Aika *et al.*, 2012; Yara Italia S.p.A, 2016).

The alarm database consists of alarm data collected during an observation period of more than four months. Each row of the database represents an alarm event (26,473 observations in total), and each column (thirty-six in total) represents a piece of information about the alarm (i.e. an “attribute”). A list of the most meaningful attributes is presented in Table 1.

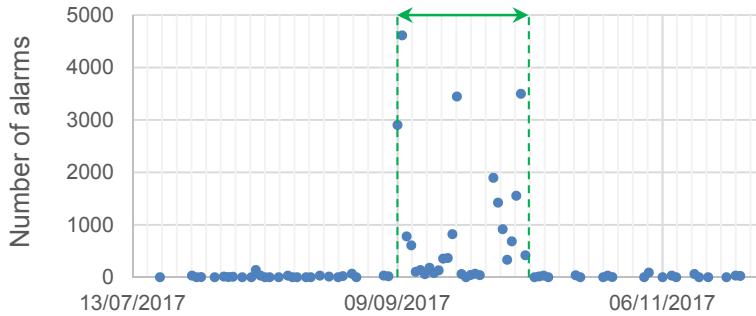
Table 1 - Alarm database attributes

Attribute	Meaning
Time Stamp	Date and time (GMT) of the alarm event.
Source	The source that triggered the alarm. It might be a measuring instrument or a PLC function.
Jxxx	The safety interlock logic associated with the alarm.
Message	The message that is shown to the operator contains the following five attributes: <ol style="list-style-type: none"> 1. the Source; 2. a concise description of the equipment involved; 3. the safety interlock logic (Jxxx); 4. the value and units of measures of the process variable; 5. the Alarm Identifier (e.g. HHH, HTRP, LLL, LTRP, ACK, etc.)
Active Time	Date and time (GMT) of the first alarm occurrence.
Data Value	The value of the process variable.
Eng. Unit	The units of measure of the process variable.

The *Alarm Identifier* (point 5. of the “Message” attribute) is a code that defines the alarm status. Examples of *Alarm Identifiers* are “HHH” (which means that the measured variable has exceeded the “high level” setpoint), “HTRP” (the measured variable has exceeded the “very high level” alarm setpoint and automatic block intervention procedures might be triggered), “IOP” (which indicates an instrumental failure or out-of-range measure), “LLL” and “LTRP” (same as “HHH” and “HTRP” but referring to a “low/very low level”).

According to Kondaveeti *et al.* (2010), an alarm event is uniquely identified by three attributes only: *Time Stamp, Source, and Alarm Identifier*.

The combination of a “source” and an “alarm identifier” is called a “unique alarm”. The time-distribution of the alarms has been assessed and represented in Figure 1.



Through the steps above, each of the first $n - 1$ occurrences of the *unique alarm* of concern is associated with a Dynamic Chattering Index (the last occurrence is excluded from the calculation). Then, the procedure is repeated for each unique alarm. The Dynamic Chattering Index assumes values between 0 and 1. The larger the index (i.e. the closer to 1), the higher the alarm chatter within one hour. According to Kondaveeti *et al.* (2013), an index value equal to 0.05 has been used as a threshold to categorise alarms into "Chattering" and "Not Chattering"; if an alarm event has $\psi_D \geq 0.05$, the alarm will show chattering in an hour.

3.2 Machine Learning simulations

A Deep Neural Network (DNN) has been trained and evaluated against the ability to predict alarm chatter. Specifically, the purpose of the algorithm is to classify alarms into two categories: "Chattering within one hour" or "Not Chattering within one hour". A database has been created containing both *features* (i.e. meaningful attributes of an alarm event) and *labels* (i.e. values or categories that the model must predict). Each row of the database represents an alarm event. The first thirteen columns represent an attribute of the alarm (i.e. a *feature*), the fourteenth column contains the *labels* associated with each alarm event. A *label* can be either "1" if the alarm will show chattering within one hour (i.e. $\psi_D \geq 0.05$) or 0 if the alarm will not show chattering within one hour (i.e. $\psi_D < 0.05$). The *features* are presented in Table 2.

Table 2 – Alarm's features

Attribute	Meaning
Y, M, d, H, m, S	Year, Month, Day, Hour, ..., Second of the alarm event
SO	The alarm Source
ID	The alarm Identifier
CN	The alarm Condition Name (i.e. the alarm <i>identifier</i> of the original alarm from the same Source)
JX	The safety interlock logic associated with the alarm
ATD	Time between the alarm event and its recovery
VAL	The value of the process variable
UNI	The units of measure of the process variable

Next, the database has been shuffled (i.e. rows have been randomly rearranged to improve data distribution) and divided in two, to obtain two distinct databases: the first database (i.e. the *training database*) comprises $\frac{3}{4}$ of the original database, the remaining part constitutes the second database (i.e. the *evaluation database*). Finally, the *label/s* have been removed from the evaluation database.

The databases have been used to *train* and *evaluate* the Deep Neural Network, whose generic architecture is shown in Figure 2.

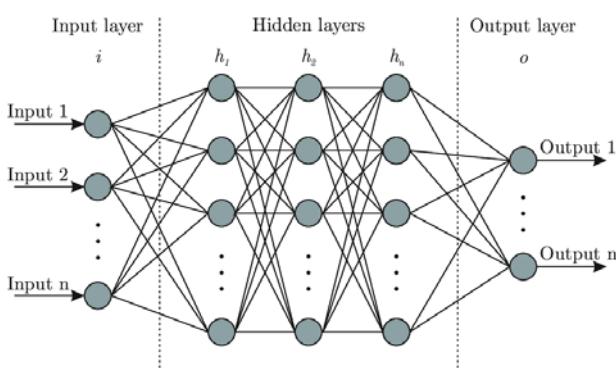


Figure 2 - Artificial neural network architecture (Bre *et al.*, 2018)

During the *training* phase, the algorithm receives as an input both the *features* (Input in Figure 2) and the associated *labels* (Output in Figure 2). During the process, the features are linearly combined and converted through non-linear functions (i.e. activation functions) into *derived features* (i.e. *hidden units*; h_1, h_2, h_n in Figure 2), which constitute the *hidden layer* of the Neural Network (Hastie *et al.*, 2009). ReLU rectifier has been used as an activation function in this work. The weights of the functions are optimised to best represent the relationship between *features* and *labels* (Hastie *et al.*, 2009). Adagrad optimiser has been used for this purpose.

The Deep Neural Network used in this work has three hidden layers with 1024, 512 and 256 hidden units, respectively. After the training, the algorithm is evaluated against the ability to predict the labels of the data included in the evaluation database (i.e. to predict the labels of alarm events that the algorithm has never “seen” before). The Machine Learning algorithm has been developed using TensorFlow r1.15.

4. Results

An example of the results obtained through the Dynamic Chattering Index approach is displayed in Table 3.

Table 3 – Dynamic Chattering Indices for FI227A LLL (Reduced version)

Time Stamps	FI227A LLL	Ψ_D (FI227A LLL)
...
2017-09-09 16:18:09	1	0.072
2017-09-09 16:18:11	1	0.071
2017-09-09 16:24:01	1	0.051
2017-09-09 16:24:03	1	0.018
2017-09-09 16:24:47	1	0.012
...

Specifically, the table includes a small portion of the Dynamic Chattering Indices related to the unique alarm FI227A LLL. The alarm warns that the flow indicator FI227A has measured a value lower than the “low level” setpoint. The first two columns of the table are the binary representation of the unique alarm (zeroes have been removed from the binary sequence for visualisation purposes). The last column of the table contains the Dynamic Chattering Indices associated with each of the alarm events. The first three indices (marked in red) indicate that the alarm will show chattering behaviour within one hour after the alarm occurrence.

The results of the Machine Learning simulation are shown in the Confusion Matrix displayed in Figure 3.

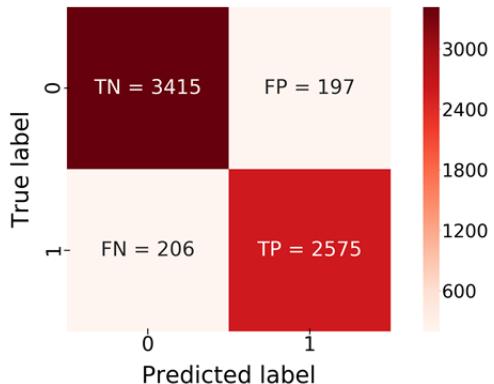


Figure 3 – DNN simulation Confusion Matrix

The metrics “TN” (i.e. True Negative) and “TP” (i.e. True Positive) together represent the number of correct predictions. “FP” (i.e. False Positive) and “FN” (i.e. False Negative) represent the number of wrong predictions. The total number of predictions can be obtained by summing all the metrics discussed above. Therefore, the algorithm produced 6393 predictions (i.e. number of alarm events in the evaluation database); 5990 of them were correct while 403 were incorrect. Besides, three additional metrics have been calculated:

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} = \mathbf{0.937} \quad (3)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} = \mathbf{0.929} \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} = \mathbf{0.926} \quad (5)$$

The *Accuracy* is the ratio between the correct predictions and the total number of predictions. The *Precision* is the fraction of correct positive predictions (i.e. predicted label = 1 and true label = 1). The *Recall* is the fraction of real positive correctly predicted. *Accuracy*, *Precision* and *Recall* are bounded between 0 and 1; the closer to 1, the better the algorithm performance.

5. Discussion

5.1 Dynamic Chattering Index

The Dynamic Chattering Index evaluates the likelihood of alarm chatter within a defined time interval (e.g. 1 hour). The method produces coherent results in most applications, but it may behave unexpectedly when few alarms occur within the time interval. Specifically, the index is sensitive to the combination of high probability and short run-lengths, a situation that may arise when few alarms occur in fast sequence within the time interval (Tamascelli, 2020). In these situations, just a couple of alarms with run-length less than 5 s could be enough to produce an index greater than 0.05 (i.e. chattering). Therefore, future research will be devoted to the development of a more reliable method for the dynamic quantification of alarm chatter.

5.2 Machine Learning simulations

The DNN model reveals excellent prediction capability. More than 93 % of the total predictions were correct, and more than 92 % of the chattering alarms were correctly identified. Despite the remarkable performance, the Deep Neural Network has not been optimised. For instance, future research will certainly investigate whether the use of a different set of *features*, as well as a different optimiser or a different set of hyperparameters (e.g. the number of hidden units), may lead to better results. As a long-term objective, future research will be devoted to the development of a method to integrate the Machine Learning model on a real industrial alarm system.

6. Conclusions

A method for Dynamic chattering assessment has been developed and the results have been used to train and evaluate a Deep Neural Network. The model has been tested against the ability to predict alarm chatter. Good results have been obtained using a “standard” model (i.e. not optimized). As previously argued, Poor alarm rationalization, chattering and alarm floods are common issues in chemical plants. In this context, Machine Learning models may meet the need for flexible, dynamic and Industry 4.0 oriented tools. Currently, chattering alarms are only addressed retrospectively; existing techniques can identify past alarm chatter but cannot predict future chattering based on actual plant conditions. Instead, the Machine Learning approach described in this work suggests that past alarm data can be used to extract knowledge and to predict alarms behaviour. These advanced models might be valuable tools in supporting the operator response during critical events.

References

- Aika K., Christiansen L. J., Dybkjaer I., Hansen J. B., Nielsen P. E. H., Nielsen A., Stoltze P., Tamaru K. , 2012, Ammonia: catalysis and manufacture. Springer Science & Business Media.
- ANSI/ISA , 2016, ‘ANSI/ISA-18.2-2016 Management of Alarm Systems for the Process Industries’, ANSI/ISA.
- Balasko B., Abonyi J. , 2007, ‘What Happens to Process Data in Chemical Industry? From Source to Applications – An Overview’, Hungarian Journal of Industrial Chemistry, 35, pp. 75–84. doi: 10.1515/133.
- Bre F., Gimenez J. M., Fachinotti V. D. , 2018, ‘Prediction of wind pressure coefficients on building surfaces using artificial neural networks’, Energy and Buildings, 158, pp. 1429–1441. doi: 10.1016/j.enbuild.2017.11.045.
- EEMUA , 2013, ‘EEMUA Publication 191 Alarm systems - a guide to design, management and procurement’.
- Hastie T., Friedman R., Tibshirani J. , 2009, The Elements of Statistical Learning. Springer-Verlag New York. doi: 10.1007/978-0-387-84858-7.
- Katzel J. , 2007, Control Engineering | Managing Alarms. Available at: www.controleng.com/articles/managing-alarms (Accessed: 23 January 2020).
- Kondaveeti S. R., Izadi I., Shah S. L., Black T. , 2010, ‘Graphical representation of industrial alarm data’, IFAC Proceedings Volumes. IFAC, 11(PART 1), pp. 181–186. doi: 10.3182/20100831-4-fr-2021.00033.
- Kondaveeti S. R., Izadi I., Shah S. L., Shook D. S., Kadali R., Chen T. , 2013, ‘Quantification of alarm chatter based on run length distributions’, Chemical Engineering Research and Design. Institution of Chemical Engineers, 91(12), pp. 2550–2558. doi: 10.1016/j.cherd.2013.02.028.
- Liu J., Kong X., Xia F., Bai X., Wang L., Qing Q., Lee I. , 2018, ‘Artificial intelligence in the 21st century’, IEEE Access, 6(April), pp. 34403–34421. doi: 10.1109/ACCESS.2018.2819688.
- Shaw J. A. , 1993, ‘DCS-based alarms: Integrating traditional functions into modern technology’, ISA Transactions, 32(2), pp. 177–181. doi: 10.1016/0019-0578(93)90039-Y.
- Tamascelli N. , 2020, A Machine Learning Approach to Predict Chattering Alarms. University of Bologna - NTNU.
- Topsoe.com , 2020, Ammonia. Available at: www.topsoe.com/processes/ammonia (Accessed: 4 April 2020).
- Yara Italia S.p.A , 2016, Relazione di riferimento della Yara Italia S.p.A. dello stabilimento di Ferrara. Available at: va.minambiente.it/it-IT/Oggetti/Documentazione/1905/10478.