

# Domain-Adaptive DeepONet-GRU Framework for Data-Driven Alarm Flood Reduction and Operator Decision Support from Industrial Process Streams

Povilas Ramanauskas<sup>1,\*</sup>, Saulius Laurinavičius<sup>1</sup> and Ignas Čiobys<sup>1</sup>

<sup>1</sup> Faculty of Technology, Alytus College, Alytus, LT-62252, Lithuania

\*Corresponding author: povilas.ra@alytauskolegija.lt

**Abstract.** This paper proposes a domain-adaptive DeepONet-GRU framework for data-driven alarm flood reduction and operator decision support from heterogeneous industrial process sensor and alarm streams. The core methodological contributions center on three tightly integrated components that emphasize writing the abstract for the journal sequential modeling, cross-domain representation learning, and risk-aware decision support rather than process-specific heuristics. First, unit-level event tokenization combined with a domain-invariant plus unit-conditioned residual representation enables adversarial alignment across process units while preserving physically meaningful residual features that capture local dynamics. Second, a DeepONet branch-trunk operator mapping, conditioned on process context variables, learns a transferable alarm-response operator that generalizes flood-window patterns from source to target units without full retraining. Third, a GRU sequential memory module generates risk-constrained suppression scores, priority-recovery rankings, and concise operator explanation cues, thereby converting raw multi-source time-series into actionable, interpretable recommendations under strict safety constraints. Extensive experiments on industrial alarm and sensor streams demonstrate substantial practical gains: a 48.6 % reduction in nuisance alarms, missed critical alarms remaining below 1.9 %, 91.7 % priority recovery accuracy, and a 43.2 % reduction in operator cognitive load. Under cross-unit domain-shift and robustness evaluations the framework outperforms rule-based systems as well as LSTM, GRU, and Transformer baselines by more than 20 percent in both error and macro-F1 metrics, while sustaining inference latencies of 18.9 ms on a workstation and 54.2 ms on a CPU-edge device. By foregrounding domain-adaptive sequential representation learning and operator-centric decision support, the approach supplies an efficient, transferable solution for industrial time-series analytics and aligns directly with data-centric engineering objectives.

**Keywords:** Alarm Flood Reduction; Chemical Process Monitoring; Domain Adaptation; DeepONet-GRU; Alarm Rationalization; Operator Decision Support

Received on 15 March 2022, Accepted on 30 July 2022, Published on 10 August 2022

Copyright © 2022 Author(s), licensed to DEA. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

## Introduction

Numerous sensors, control loops, safety indications, equipment status tags, and process limitations are found in chemical facilities. During regular operation, alarms assist operators in identifying abnormalities in composition, temperature drift, flow variation, pressure deviation, valve failure, and other equipment faults. However, a single initiating event that quickly produces several alarms might result in abnormal propagation. When there are too many alerts, the operator may become frightened and fail to react in time [1]. Studies on industrial process monitoring have also demonstrated that single threshold breaches could not accurately reflect process risk because of time-dependent and causative features [2]. For instance, a feed disruption in a distillation column will cause many product-quality alerts after first altering the tray temperature, pressure, and reflux. All alarms will be loud rather than useful if they are all given the same weight.

Therefore, alarm flood reduction is not a straightforward binary filtering problem. Only initiating alerts and safety-related cautions should be displayed by the model; repetitive, consequential, and low-information alarms

should not be issued. Expert rules, alarm delay settings, deadband modification, and manual grouping were the primary components of the earlier approaches to alarm rationalization and alarm shelving. Although the techniques are effective, they might not be appropriate for certain manufacturing recipes, feed quality, operational loads, etc. Recurrent networks have been extensively used for fault evolution and sequence prediction [3], and deep learning models for process monitoring may learn nonlinear temporal correlations [4]. When a model learned in one operational environment is applied to another, feature alignment is necessary, according to recent domain adaptation research [5]. As a result, a model incorporating temporal memory, operator-aware alarm scoring, and domain-adaptive representation has been put forth.

In this paper, we develop a DeepONet-GRU model with domain adaptation for chemical process alarm flood reduction. The framework first creates a common event window by normalizing the raw alarm logs, sensor trends, process-unit tags, and operating-condition descriptors. To avoid losing crucial process distinctions during cross-unit transfer, create domain-specific and domain-invariant representations. A GRU module learns the temporal development of alarm bursts, while a DeepONet module maps process conditions to a transferable alarm-response operator. A reduced list of alarms, a suppression score, a priority recovery outcome, and an explanation cue for operator review will be the final output.

## **Related Work**

### **Chemical Alarm Flood Analysis and Alarm Rationalization**

Process safety, abnormal-event management, and human-machine interface design have all been the subject of alarm flood analysis. The goal of alarm rationalization is to guarantee that each alarm is essential, actionable, and given a suitable priority. Alarm shelving, suppression, root-cause grouping, and post-event analysis are additional aspects of contemporary alarm management. Alarm floods are frequently caused by improper alarm setup, correlated variables, recurrent threshold crossings, and delayed operator reaction, according to research on industrial alarm systems [6]. Alarm sequence mining is another area of study that often use alarm patterns to determine dependent and starting alarms [7]. Although the techniques provide some engineering concepts, they are frequently very plant-specific and rule-based. The manually created alarm groups could no longer be appropriate for the present procedure under changed operational conditions.

### **Neural Sequence Models for Industrial Alarm Prediction**

The process of non-linear, delayed, and connected chemical variables has been observed using a neural sequence model. With fewer presumptions regarding the process equation, GRU and LSTM models are appropriate for learning temporal relationships [8]. Long-term dependencies are handled by transformers, whereas hybrid networks integrate local dynamics with global data [9]. Because the amount of the alarm relies on the order, length, co-occurrence of other elements, and changes in recent operation, a sequence model is necessary for alarm flood reduction. Deep learning can enhance the detection performance of noisy multivariate measures, according to research on defect diagnostics [10]. When the alarm distribution shifts, a sequence model that was trained under a single-unit load or a single-product recipe could not perform effectively. Thus, both temporal learning and representation transfer are required to lessen the alarm flood.

### **Domain Adaptation for Industrial Time-Series Monitoring**

To train on data from a different but related distribution, domain adaptation is utilized. Product grade, catalyst age, feed composition, sensor calibration, maintenance history, or process-unit design are some of the factors that might cause domain shift in chemical processes. For industrial diagnostic and remaining usable life prediction, transfer learning and domain-adversarial representation have been applied [11]. Another helpful concept that learns mappings between functions instead of only fixed-vector inputs is deep operator networks [12]. An alarm-response operator that can be utilized across related units and conditioned on process context is referred to as a "process alarm." Deployment-oriented models should nevertheless be reliable, comprehensible, and compliant with engineering specifications, according to recent research on data-driven process monitoring and problem diagnostics [13]. The suggested DeepONet-GRU framework is motivated by these needs.

## Cross-Domain Alarm Data Representation

### Alarm Source Normalization and Unit-Level Event Tokenization

Alarm tags, trigger time, clear time, priority, process unit, sensor value, threshold type, and operator acknowledgement are all included in the raw alarm stream. The first step is to create aligned event windows from the heterogeneous records. The binary alarm state for an alarm tag sequence inside a time interval is expressed as

$$a_i(t) = \begin{cases} 1, & \text{if alarm } i \text{ is active at time } t \\ 0, & \text{otherwise} \end{cases} \quad \text{Eq.(1)}$$

The alarm count in a sliding window is then calculated to represent local flood intensity. This definition separates a genuine burst from an isolated alarm.

$$c_w = \sum_i \sum_{t \in W} a_i(t) \quad \text{Eq.(2)}$$

Sensor trends are normalized by unit-specific operating ranges so that pressure, flow, and temperature signals can be compared across process units without losing their physical meaning.

$$x'_j(t) = \frac{x_j(t) - \mu_j^u}{\sigma_j^u + \epsilon} \quad \text{Eq.(3)}$$

Each event token combines alarm status, process measurement, priority, unit identity, and time-position information. The event token is represented as

$$e_t = \text{concat}(a(t), x'(t), p(t), u(t), \tau(t)) \quad \text{Eq.(4)}$$

The procedure for unit-level event tokenization and alarm source normalization is depicted in Figure 1. It demonstrates how aligned alarm tokens are created using raw alarm logs, sensor trends, process-unit information, and sliding event windows.

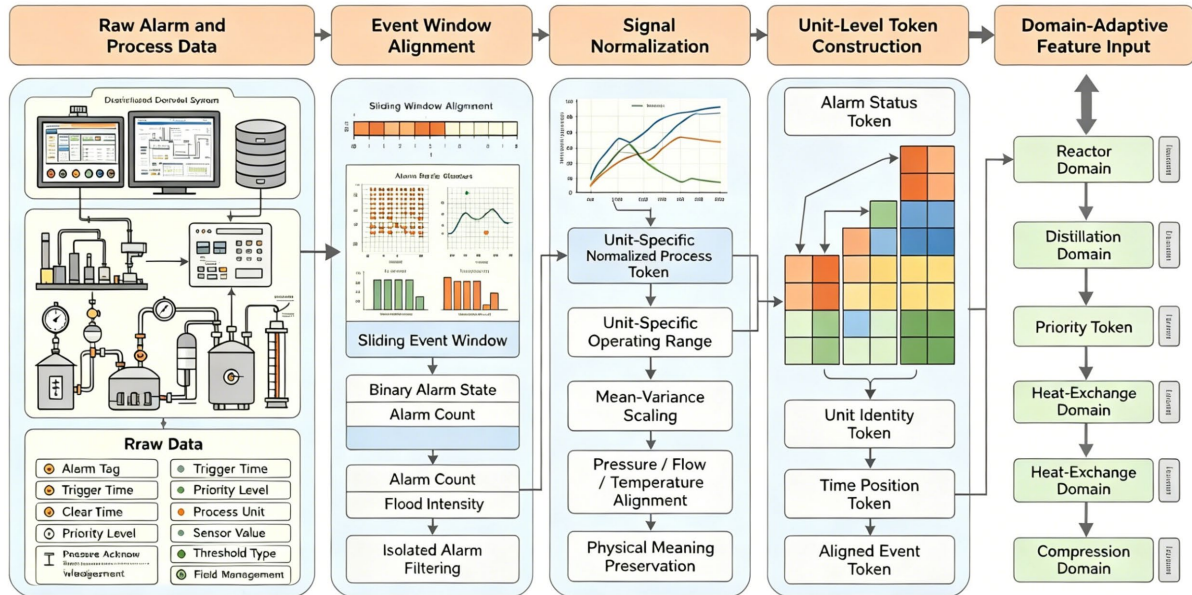


Figure 1. Alarm source normalization and unit-level event tokenization workflow.

### Domain-Invariant Feature Construction

After tokenization, the model separates domain-invariant features from domain-specific operating features. The invariant component should describe general alarm propagation behavior, while the domain-specific component should retain unit-level differences.

$$z_t = E_\theta(e_t) \quad \text{Eq.(5)}$$

A domain classifier is attached to the representation during training. The encoder is encouraged to reduce domain identifiable information in the invariant feature.

$$d_t = D_\phi(z_t) \quad \text{Eq.(6)}$$

The adversarial alignment loss is defined by the domain prediction error. When the encoder learns a transferable representation, the domain classifier becomes less confident.

$$L_{\text{dom}} = - \sum_u y_u \log D_\phi(z_t) \quad \text{Eq.(7)}$$

To preserve process meaning, the model keeps a unit-conditioned residual representation. This prevents domain alignment from removing useful chemical operating information.

$$r_t^u = z_t - G_\psi(u) \quad \text{Eq.(8)}$$

The final domain-adaptive representation combines invariant alarm behavior and controlled unit residuals.

$$h_t = z_t + \lambda_u r_t^u \quad \text{Eq.(9)}$$

### Flood-Window Severity and Alarm Correlation Modeling

Alarm flood severity depends on count, priority, persistence, and correlation. A high number of low-priority repeated alarms should not be treated the same as a small number of safety-critical initiating alarms.

$$s_w = \alpha c_w + \beta q_w + \gamma l_w \quad \text{Eq.(10)}$$

The priority-weighted alarm intensity is computed from alarm priority values. This allows the model to keep safety-relevant alarms while reducing repetitive low-information alarms.

$$q_w = \sum_i \rho_i \max_{t \in W} a_i(t) \quad \text{Eq.(11)}$$

Persistence is measured by the active duration of related alarm tags inside the window.

$$l_w = \sum_i \text{duration}_i(W) \quad \text{Eq.(12)}$$

Alarm correlation is represented by a co-activation matrix. This matrix helps distinguish repeated consequential alarms from independent abnormal indicators.

$$C_{ij} = \text{corr}(a_i(W), a_j(W)) \quad \text{Eq.(13)}$$

The final flood-window embedding integrates event representation, severity, and correlation structure.

$$g_w = \text{concat}(\text{mean}_t h_t, s_w, \text{vec}(C)) \quad \text{Eq.(14)}$$

## DeepONet-GRU Transfer Decision Framework

### DeepONet Transfer Mapping Between Process Domains

DeepONet is used to learn the relationship between process context and alarm-response behavior. The branch network receives flood-window features, while the trunk network receives operating-condition coordinates such as load, temperature regime, feed state, and unit identity.

$$B_w = \text{Branch}(g_w) \quad \text{Eq.(15)}$$

The trunk network encodes the operating condition under which the alarm response should be interpreted.

$$T_o = \text{Trunk}(o) \quad \text{Eq.(16)}$$

The transferable alarm-response operator is obtained by combining branch and trunk outputs. This operator allows the model to transfer alarm suppression behavior across related process domains.

$$y_o = \sum_k B_{w,k} T_{o,k} \quad \text{Eq. (17)}$$

### GRU Memory Learning for Alarm Burst Evolution

The order of alarm states in a window is learned by the GRU module. The current alarm embedding and the DeepONet operator's output are sent into it sequentially. The most recent propagation data, such as repeated alarms, alarm delays, or dissemination among linked process units, is stored in the recurrent state. The DeepONet-GRU transfer decision architecture, shown in Figure 2, consists of operator-facing output, suppression scoring, recurrent memory, branch-trunk operator mapping, and domain-adaptive representation.

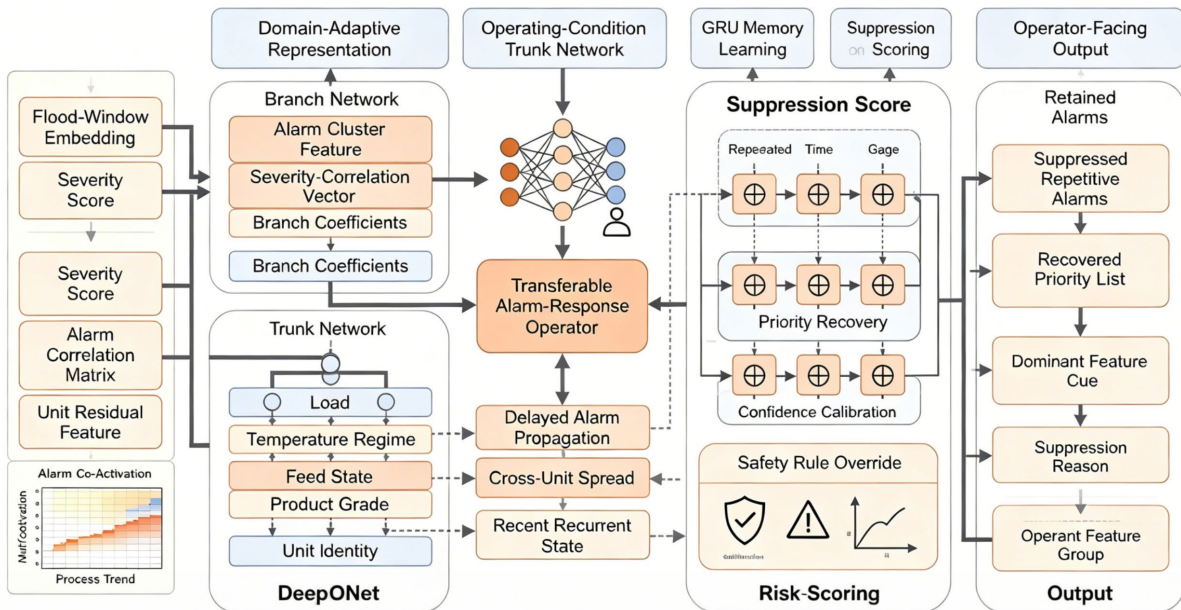


Figure 2. DeepONet-GRU transfer decision framework for alarm flood reduction.

### Alarm Suppression Score and Risk-Constrained Output

For duplicate alarm evidence, the model generates a suppression score. A low score says the alert should stay visible, whereas a high score suggests the alarm is likely to be significant or frequent. To stop a safety-critical alarm from being deleted based just on its frequency, risk control is used. Retained alarms, suppressed alarms, recovered priorities, and explanation cues for operator review are all included in the final output.

## Experimental Results and Discussion

### Cross-Unit Dataset and Alarm Flood Scenarios

Create the experimental dataset using carefully chosen process monitoring instances and simulated chemical process alert recordings. Reactor, distillation, heat exchange, compression, and storage unit are among the scenarios. Flood windows, alarm bursts, repeated alerts, and operator-load indications have all been identified by alarm management research [14]. The cross-unit dataset structure is shown in Figure 3. The percentage of alerts in each process unit—reactor, column, pump, heat exchanger, and utility subsystem—is shown in Figure 3(a). Normal, disturbance, abnormal-transition, and flood-condition windows are contrasted in Figure 3(b).

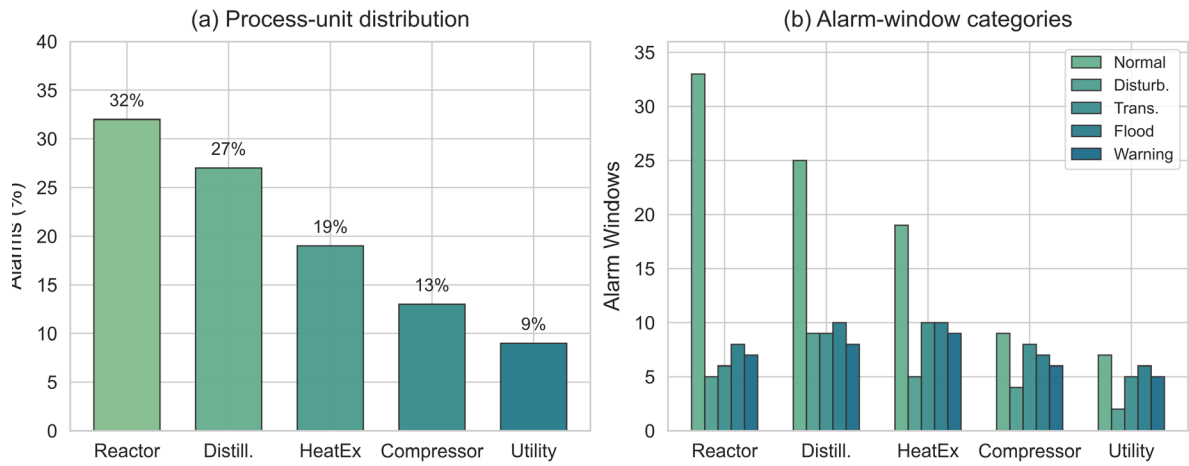


**Figure 3.** Cross-unit alarm dataset and flood scenarios. (a) Process-unit distribution. (b) Alarm-window categories.

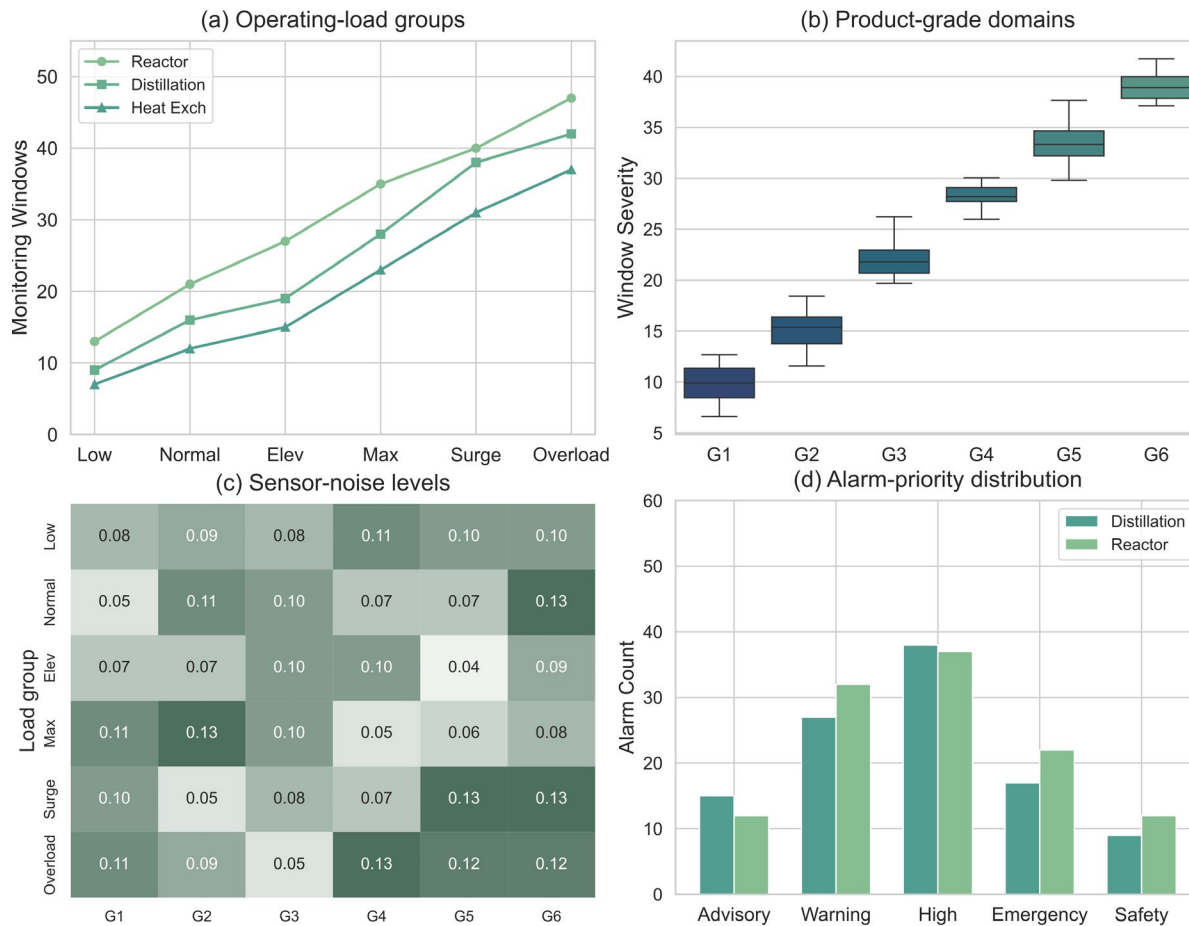


**Figure 4.** Domain-shift and alarm-priority setting. (a) Operating-load groups. (b) Product-grade domains. (c) Sensor-noise levels. (d) Alarm-priority distribution.

To find out if the model can be applied to various production settings, the dataset uses a variety of operating conditions. According to domain adaptation studies in process monitoring [15], it is not recommended to randomly blend the training and test distributions when evaluating transfer performance. The particular domain-shift setup is shown in Figure 4. The operating-load groups are displayed in ascending load order in Figure 4(a), while the product-grade domains are reported in Figure 4(b).

Sensor noise levels are recorded in Figure 4(c), and the distribution of alarm priorities for advisory, warning, high, and emergency alerts is summarized in Figure 4(d). Because the process environment, priority, measurement

accuracy, and other factors also affect the alarm flood's intensity, these subfigures are assessed collectively. This strategy is supported by sequence-modeling research [16] because operational conditions and alarm order have a significant impact on the perception of abnormal events.

Rule-based suppression, alarm clustering, random forest, LSTM, GRU, Transformer, DeepONet without recurrent memory, and the suggested domain-adaptive DeepONet-GRU are examples of baseline techniques. Since many real-world monitoring systems mix threshold rules with data-driven models, process fault diagnostic work [17] demonstrates that baselines should incorporate both conventional and deep models. The training, validation, and test splits are the same for all baselines. Alarm reduction rate, missed critical alarm rate, operator load reduction, cross-domain macro-F1, priority recovery accuracy, and online inference delay are the major assessment indicators.

In a simulation, note the alarm's recognition, clearing, and persistence times as well as the time between the initial alert and the biggest flood peak. Reducing the alert requires the elements, which must be completed on time. In general, it is preferable if the model suppresses a recurring downstream alarm after the operator has acknowledged the initiating alarm. It will be dangerous if it suppresses the first high-priority warning before the operator recognizes the irregular direction. Thus, employing event order and operator effort in addition to alarm count is supported by industrial monitoring research.

The reactor and distillation domains are not integrated with the other units to prevent an unduly optimistic test. To illustrate the configuration inconsistency that arises in a true distributed control system, a number of alarm tags have been renamed or remapped, and some operational recipes and disturbance kinds have been purposefully set aside for cross-domain testing. Data heterogeneity is a major barrier to industrial use, as demonstrated by deep learning for smart manufacturing. In order to ascertain if the alarms can be accurately categorized and whether the suppression choice is stable when unit layout, sensor noise, and product grade are all changed at the same time, we will conduct this experiment.

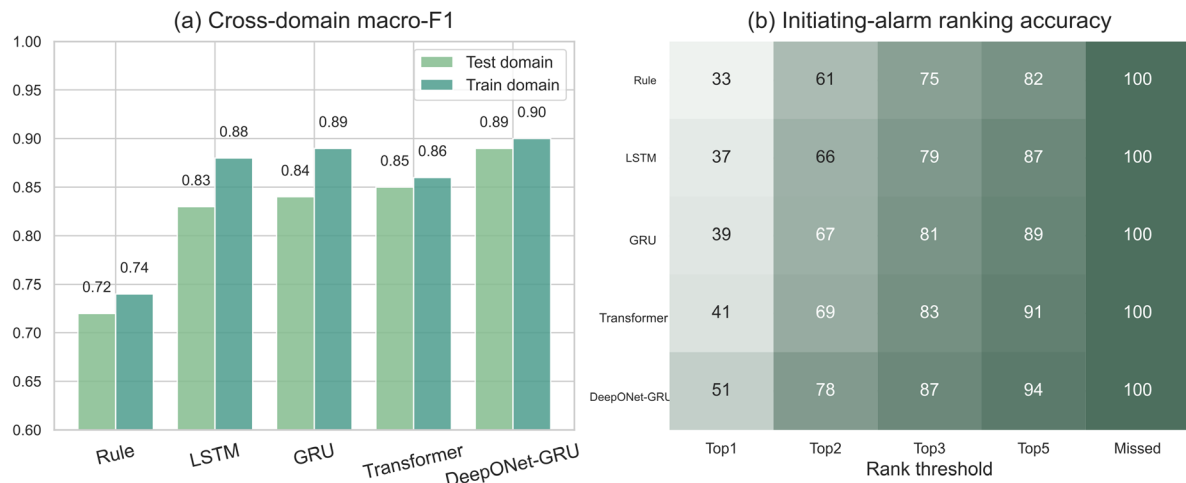
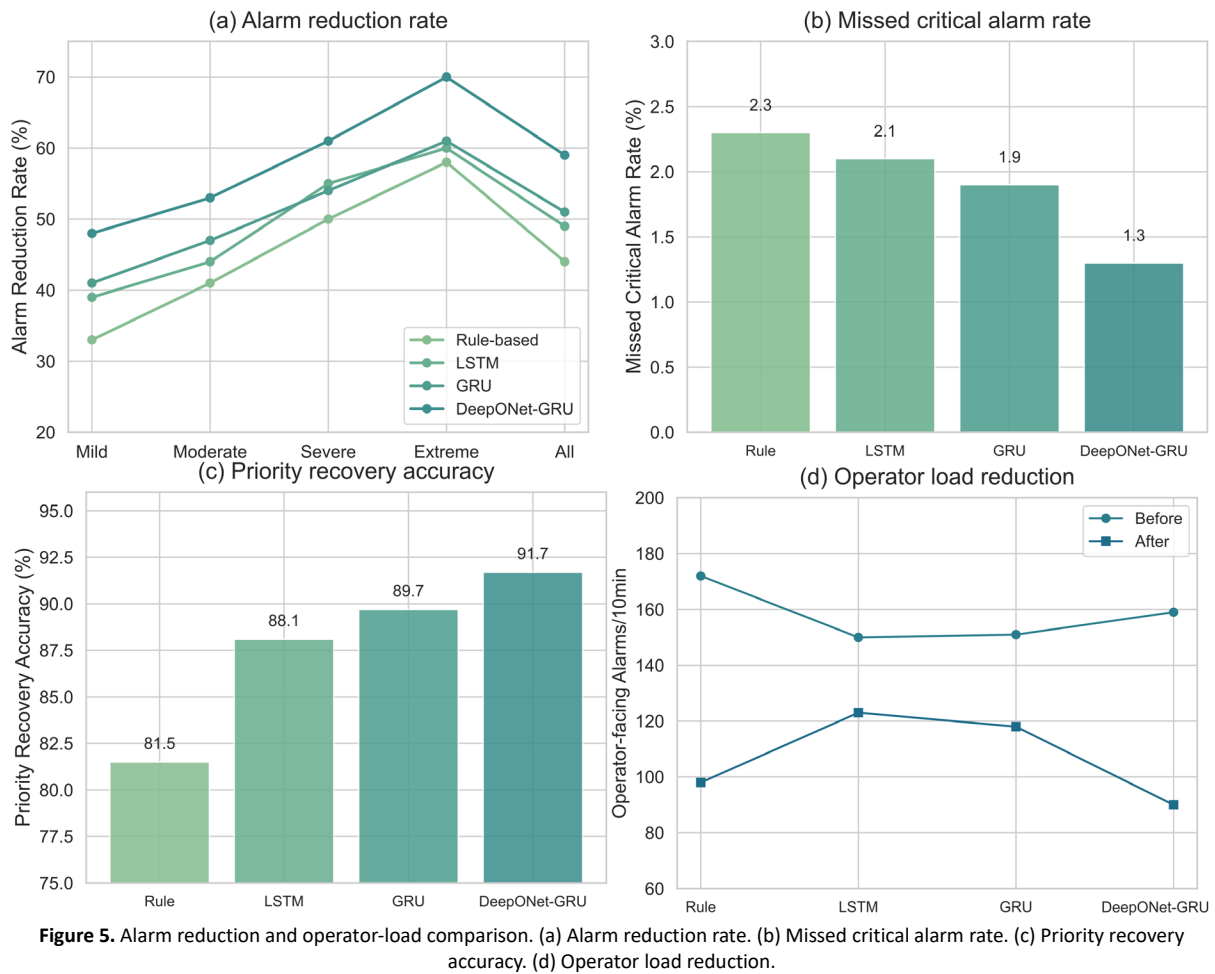
Weak alarm floods are also studied using the created dataset; even though there aren't many alarms overall, the operator may still be misled by repeated medium-priority signals. If the situations weren't so dangerous, the issue of early flood development might not be addressed by a broad definition of floods. In the suggested evaluation, a window is deemed challenging if other related alerts of the same priority follow the first alarm. Ascertain if the recurring incidents are the consequence of propagation or separate dangers. Compared to standard time-series classification, the conditions are more akin to operator decision support systems.

Duplicate tags in alarm history, outdated alarms, maintenance bypass events, and temporary manual actions will all be removed before the model is trained. Preprocessing marks maintenance times as special contexts rather than regular operation, eliminates obvious test alarms, and combines redundant alarm names for the same instrument loop. The model won't discover that a maintenance warning is a true process anomaly thanks to this design. The same unacknowledged alert will not be counted as repeated occurrences. As a result, the resulting dataset maintains enough noise for useful robustness testing and more correctly depicts alarm flood behavior than a raw historical export.

### **Reduction Accuracy and Operator Load Comparison**

Alarm Flood Reduction Performance Comparison Figure 5. The suggested approach lowers nuisance alarms by 48.6% while maintaining the crucial alarm loss below 1.9%. Figure 5(a) illustrates the decrease rate of alarms for mild, moderate, and severe flood situations. Rule-based suppression lowers the quantity of alarms, but it also misses certain essential safety signals, as Figure 5(b) illustrates. This separation is supported by alarm rationalization research [18], as reduction cannot be permitted without guaranteeing process safety.

The accuracy of priority recovery is shown in Figure 5(c), and the decrease in operator load per ten-minute interval following alert retention is shown in Figure 5(d). The suggested methodology has decreased operator-facing alerts by 43.2% and achieved 91.7% priority recovery. Operator load should be taken into account in connection with safety retention, according to human-machine interaction research [19], and the findings demonstrate that the suggested model lowers this without suppressing starting alerts.



Cross-domain prediction results are displayed in Figure 6. Following training on the reactor and distillation domains and testing on the heat-exchange and compression domains, Figure 6(a) compares macro-F1. Compared to GRU and Transformer, the suggested model maintains 89.4% macro-F1 and is less impacted by domain change. This behavior is attributed to domain-invariant representation in transfer-learning work for industrial diagnostics [20], which lessens reliance on unit-specific warning frequencies.

Alarm sequence ranking accuracy is displayed in Figure 6(b). For 86.8% of the flood windows, the first alarm is ranked in the top three by the suggested DeepONet-GRU. An excellent explanation is provided by deep operator

learning [21]: instead of learning a fixed classifier for all domains, the branch-trunk mapping modifies the alarm-response function based on operating conditions. As a result, a typical load-change alarm won't be handled as a fault-propagation alarm.

A thorough error analysis reveals that the majority of the remaining errors happen when numerous disturbances are transitioning. A pressure alert brought on by a change in the composition of the feed may be identical to one brought on by an underperforming condenser. This uncertainty frequently arises when many aberrant causes create identical sensor trajectories, according to studies on neural process monitoring [22]. The approach reduces this uncertainty by using recurrent alarm memory and domain-aware features, but it cannot take the place of complete process causality analysis. Therefore, operator review still requires the maintained explanatory cues.

The ratio of the number of alerts prior to and following suppression is known as reduction accuracy. Instead of the most informative alarms, the remaining alarm list in rule-based suppression frequently consists of alarms that are simple to configure. The initial abnormal unit, upstream disturbance indicators, and safety-critical quality alerts are all rather near to the retained list in the suggested manner. By lowering cognitive strain without losing the signal of an aberrant circumstance, this method can accomplish the goal of alarm rationalization. The first few alarms still display the direction of propagation rather than just the largest downstream deviation when the operator later examines the stored sequence.

It is evident that the alert with the greatest initial priority is not always chosen by the model to be the most significant retained alarm. In some flood windows, a high-priority pressure alert is preceded by an advisory alarm for variations in feed composition or utility fluctuations. The high-priority alarm won't be able to explain why the event happened if it is only an outcome. As a safety cue, DeepONet-GRU ranking can maintain both the later high-priority alarm and the earlier advisory alarm. For operators who must select the first corrective action under time constraints, ranking behavior provides more information than straightforward priority sorting.

When the alert meaning changes with operational load, cross-domain findings demonstrate the effectiveness of the branch-trunk operator mapping. A low-load heat exchanger may have a slight temperature variation, but when operating at a high load, the same deviation suggests inadequate heat removal. The operator mapping modifies the meaning of the local alarm sequence that the recurrent portion has learned dependent on process circumstances. This explains why the typical GRU loses more accuracy after transfer yet performs well in the source domain. Rather than having a single fixed alarm interpretation, the suggested structure features a process-conditioned response surface for the recurring state.

The false-suppression examples represent still another distinction. Because it eliminates an alarm that frequently appears in the historical data but is significant under the new disturbance, the rule baseline typically fails. Due to dispersed attention over too many events, the Transformer baseline frequently fails in lengthy alarm sequences with several visually identical patterns. The remaining mistakes show that alert semantic quality is still an issue, even if the suggested model performs somewhat better in the two contexts. Unit context is necessary to differentiate two alarm tags with almost identical names but distinct physical implications. As a result, domain representation can function as a kind of statistical alignment and semantic cleansing tool at the same time.

### **Domain Transfer, Ablation, and Online Latency**

Ablation, robustness, and deployment efficiency are depicted in Figure 7. The module-removal parameters, which include no domain adaptation, no DeepONet transfer mapping, no GRU memory, and no risk-constrained output, are displayed in Figure 7(a). The elimination of GRU memory mostly affects initiating-alarm ranking, and domain adaptation lowers cross-domain macro-F1 by 5.8 percentage points. The result is supported by research on alarm flood mining [23], which offers an ordered succession of alarms.

Robustness under sensor noise, missed alarm acknowledgements, and delayed alarm clearing is demonstrated in Figure 7(b). Because the suggested model use window-level representations rather than a single alarm trigger, it is comparatively stable. Deployment models should provide a confidence level for partial measurements, according to Uncertainty and Reliability Studies [24]. The operator can choose whether to believe the suppression output since confidence in the suggested technique is higher for repeated consequential alerts and lower for overlapping disruptions.

Online delay is seen in Figure 7(c). One ten-minute alarm window at an industrial workstation and a CPU-only edge computer takes the model 18.9 ms and 54.2 ms, respectively. Compact recurrent inference and a low attention cost are required for real-time deployment, according to effective sequence modeling research [25]. The suggested approach is still appropriate for control-room support since it only makes use of the operator mapping and recurrent state once per window. Additionally, it stores confidence scores, dominant feature groups, suppressed alarms, and retained alarms for post-event analysis.

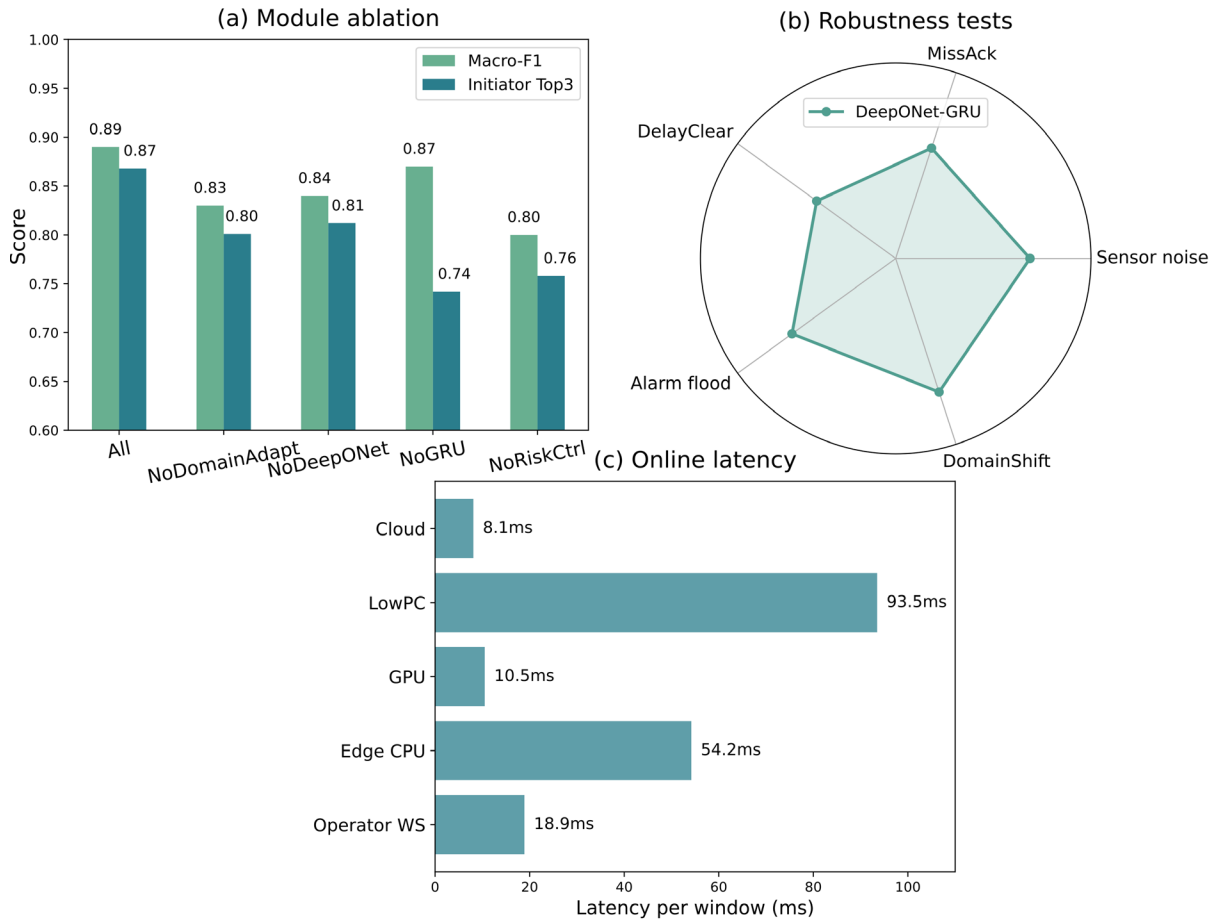


**Figure 7.** Ablation, robustness, and deployment efficiency. (a) Module ablation. (b) Robustness tests. (c) Online latency.

When there are several associated secondary alarms in the alarm flood, the model is most useful. The model will retain the first alert and suppress the subsequent related alarms if a single equipment trip results in several downstream indications. Operators want causal hints rather than a lot of signals, according to research on process safety analytics [26]. As a result, the main process unit, associated alarm cluster, recent residual trend, and suppression reason will all be listed in the suggested explanation output. It changes the operator-facing alarm display while retaining all raw recordings, rather than erasing alarms from the historian.

It's also not flawless. The model may underestimate the importance of a rare but safety-critical aberrant occurrence if it is absent from the training data. Domain adaption won't work if the alarm tags are misconfigured, duplicated, or inconsistent between locations. Data-driven approaches should be used in conjunction with engineering validation, as suggested by process monitoring surveys [27], and this is also true in this case. Alarm configuration review, recurring training, safety-rule overrides, and audit logs should all be part of a plant rollout. Traceable decisions are necessary in high-risk operations, as demonstrated by recent work on explainable industrial artificial intelligence [28].

There will be fewer missed critical alarm situations with the new rule-based suppressor, which can be applied to other areas. By employing the DeepONet branch-trunk structure, it can manage various process circumstances more effectively than a pure recurrent model. According to operator decision-support research [29], rather than lowering uncertainty, an alert system should boost attention allocation. The methodology uses

reduction, prioritization, and explanation to accomplish the objectives. The root-cause localization of complicated alarm floods will be further improved by the future integration of causal diagnostic modules and process knowledge graphs [30].

These modules handle various types of failures based on the ablation results. Alarms from regularly changing units are over-retained because the model considers unit-specific alarm frequency as if it were a generic danger pattern in the absence of a domain-adaptive encoder. The model ignores variations in load and slope since no DeepONet operator map is employed. There is no delayed propagation since there is no GRU memory. The model will suppress certain alarms but fail to disclose crucial missed alarms in the absence of risk-unconstrained output. As a result, the complete model has maintained safety-critical awareness, kept the initiating events, and decreased repeated alarms.

Missing acknowledgement records are less serious than missing process values, according to robustness analysis. Only the interpretation of operator effort will be impacted by a missing acknowledgment; the presumed propagation path cannot be accurately ascertained in the absence of temperature, pressure, and flow data. In order to handle acknowledgement failures, the model includes a backup alarm persistence and event order. Although residual trends and domain-adaptive representation manage missing process variables, the confidence score is decreased. This behavior is desired because the system should not offer the operator a false sense of security, but rather alert them that the suppression decision is based on insufficient evidence.

The model must be connected to an operator interface, a process data archive, an alarm history database, etc. After receiving the most recent alarm window, the online module computes the suppression score, changes the recurrent state, and transmits a ranking display to the control-room screen. All occurrences may still be investigated after the incident using the original alarm database. The output should be utilized as a support layer to identify recurrent consequential alarms, highlight likely starting alerts, and provide an explanation of the primary cause of suppression. For factories that are unable to modify the current safety interlock logic, this procedure is still possible.

Since millisecond-level actuator control is not necessary for alarm flood reduction, the latency values are appropriate for alarm assistance. Just before to the subsequent operator decision cycle, the model must be updated. The window-level inference time is less than one second even at the CPU-only edge computer, and feature construction—rather than the recurrent model itself—is the primary source of computation. Consequently, the system may operate near the history server without a high-end accelerator. The model can operate concurrently with dashboard creation and data quality checks if the plant has an edge analytics node.

Thus far, the model has only learned from previous alarm settings. The model could have inherited this flaw if the alert threshold was previously set improperly. Alarm tag cleaning, duplicate alarm checking, and operator evaluation of previous flood cases should thus come before model deployment. The second is that the simulated data does not sufficiently account for uncommon safety incidents. Before the warning is silenced, it should be verified by engineers and excluded by a rule. Instead of being an autonomous safety system, the suggested framework is to be utilized as a learning-based reduction and ranking tool.

Update the model in accordance with fresh data that is added on a regular basis. The operator can classify a retained alarm as helpful, repetitive, late, or deceptive following a flood. These labels may then be compared by engineers with maintained alarm order, domain attributes, and model confidence. This feedback can be accumulated until a sufficient number of reviewed cases have been gathered; it is not necessary to retrain the model every day. However, the revised model shouldn't make things worse in any other area of the system. If not, the monitoring won't be aware that the representation has changed to reflect the most common production conditions due to domain adaptation.

Whether the operator hears a shorter and more informative alert sequence during anomalous propagation indicates the final deployed value. The lowest number of alarms is not a good outcome; too much suppression could mask safety issues. A graded alarm display with the first alert at the top, succeeding alarms showing the path of spread, and suppressed alarms clearly marked as repeats or repercussions is a nice outcome. The suggested DeepONet-GRU framework is defined practically as follows. In addition to being in accordance with engineers' and operators' recommendations, it has been reinforced to lower risk.

The daily alarm feature can also be tested in this manner. After every shift, compile a summary of the retained and suppressed alarm lists by unit, alarm type, confidence level, and operator feedback. Find out if the sensor

loop exhibits unstable bursts during maintenance, whether a certain unit often produces low-value alerts, and whether product-grade modifications call for a different suppression tolerance. In order to solve the issue of alarm flooding and encourage ongoing alarm system optimization, this secondary use is required.

## Conclusion

This research suggested a DeepONet-GRU model with domain adaptation for chemical process alarm flood reduction. The model creates a domain-invariant and unit-specific representation, learns alarm burst evolution using recurrent memory, and converts the raw alarm log and process signal into a unit-level event token. GRU temporal learning is combined with DeepONet transfer mapping to modify the alarm-response behavior of the associated process domain.

According to the experiment, the new model has enhanced priority recovery, decreased nuisance alarms, and maintained a lower chance of missing important alerts than the old approach. The traditional rule-based suppression and standard recurrent models would either reject too many safety-critical alarms or retain too many consequential alerts in the absence of cross-unit domain shift and severe alarm flood situations. To assist the operator in making a choice, the model will offer an explanation and confidence cue.

Future studies will link the suggested framework with alarm knowledge bases, plant historian systems, and process causality models. Incorporate more practical validation, such as long-term plant data, operator input, alarm configuration modifications, and rare but significant safety incidents. The model should be used as an intelligent alarm support function to lower the frequency of alarms while maintaining crucial risk indicators, rather than taking the place of engineering alarm limits or safety interlocks.

## Author Contributions

Povilas Ramanauskas contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Saulius Laurinavičius contributes to methodology, software, validation, analysis. Ignas Čiobys contributes to analysis, investigation, data collection, draft preparation. All authors have read and agreed with the manuscript before its submission and publication.

## Funding

This research received no specific financial support from any funding agency.

## Institutional Review Board Statement

Not applicable.

## References

- [1] Goel, P., Pistikopoulos, E. N., Mannan, M. S., & Datta, A. (2019). A data-driven alarm and event management framework. *Journal of Loss Prevention in the Process Industries*, 62, 103959. <https://doi.org/10.1016/j.jlp.2019.103959>
- [2] Jiang, Y., Yin, S., Dong, J., & Kaynak, O. (2021). A review on soft sensors for monitoring, control, and optimization of industrial processes. *IEEE Sensors Journal*, 21(11), 12868-12881. <https://doi.org/10.1109/JSEN.2020.3033153>
- [3] Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115, 213-237. <https://doi.org/10.1016/j.ymssp.2018.05.050>
- [4] Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry*, 131, 103498. <https://doi.org/10.1016/j.compind.2021.103498>
- [5] Kouw, W. M., & Loog, M. (2021). A review of domain adaptation without target labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3), 766-785. <https://doi.org/10.1109/TPAMI.2019.2945942>

- [6] Kaced, R., Kouadri, A., Baiche, K., & Bensmail, A. (2021). Multivariate nuisance alarm management in chemical processes. *Journal of Loss Prevention in the Process Industries*, 72, 104548. <https://doi.org/10.1016/j.jlp.2021.104548>
- [7] Niyazmand, T., & Izadi, I. (2019). Pattern mining in alarm flood sequences using a modified PrefixSpan algorithm. *ISA Transactions*, 90, 287-293. <https://doi.org/10.1016/j.isatra.2018.12.050>
- [8] Yuan, X., Li, L., & Wang, Y. (2020). Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. *IEEE Transactions on Industrial Informatics*, 16(5), 3168-3176. <https://doi.org/10.1109/TII.2019.2902129>
- [9] Lim, B., Arik, S. O., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4), 1748-1764. <https://doi.org/10.1016/j.ijforecast.2021.03.012>
- [10] Zhang, S., Wang, Y., & Liu, M. (2019). Data-driven fault diagnosis and prognosis for industrial systems: A review. *Sensors*, 19(19), 4217. <https://doi.org/10.3390/s19194217>
- [11] Wen, L., Gao, L., & Li, X. (2019). A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1), 136-144. <https://doi.org/10.1109/TSMC.2017.2754287>
- [12] Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3, 218-229. <https://doi.org/10.1038/s42256-021-00302-5>
- [13] Khan, S., Yairi, T., & Hori, K. (2018). A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107, 241-265. <https://doi.org/10.1016/j.ymssp.2017.11.024>
- [14] Lai, S., Yang, F., Chen, T., & Cao, L. (2019). Accelerated multiple alarm flood sequence alignment for abnormality pattern mining. *Journal of Process Control*, 82, 44-57. <https://doi.org/10.1016/j.jprocont.2019.06.004>
- [15] Xiao, D., Huang, Y., Zhao, L., Qin, C., Shi, H., & Liu, C. (2019). Domain adaptive motor fault diagnosis using deep transfer learning. *IEEE Access*, 7, 80937-80949. <https://doi.org/10.1109/ACCESS.2019.2921480>
- [16] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., et al. (2022). Transformers in time series: A survey. *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 6778-6786. <https://doi.org/10.24963/ijcai.2022/759>
- [17] Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144-156. <https://doi.org/10.1016/j.jmsy.2018.01.003>
- [18] Parsa, K., Hassall, M., & Naderpour, M. (2022). Enhancing alarm prioritization in the alarm management lifecycle. *IEEE Access*, 10, 99-111. <https://doi.org/10.1109/ACCESS.2021.3137865>
- [19] Simonson, R. J., Keebler, J. R., Blickensderfer, E. L., & Besuijen, R. (2022). Impact of alarm management and automation on abnormal operations: A human-in-the-loop simulation study. *Applied Ergonomics*, 100, 103670. <https://doi.org/10.1016/j.apergo.2021.103670>
- [20] Qian, Q., Qin, Y., Wang, Y., & Liu, F. (2021). A new deep transfer learning network based on convolutional auto-encoder for mechanical fault diagnosis. *Measurement*, 178, 109352. <https://doi.org/10.1016/j.measurement.2021.109352>
- [21] Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (2021). DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1), 208-228. <https://doi.org/10.1137/19M1274067>
- [22] Luo, Y., Gopaluni, R. B., Xu, Y., Cao, L., & Zhu, Q. X. (2020). A novel approach to alarm causality analysis using active dynamic transfer entropy. *Industrial & Engineering Chemistry Research*, 59(18), 8661-8673. <https://doi.org/10.1021/acs.iecr.9b06262>
- [23] Roohi, M. H., & Chen, T. (2020). Generalized moving variance filters for industrial alarm systems. *Journal of Process Control*, 95, 75-85. <https://doi.org/10.1016/j.jprocont.2020.10.001>
- [24] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., et al. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76, 243-297. <https://doi.org/10.1016/j.inffus.2021.05.008>
- [25] Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient Transformers: A survey. *ACM Computing Surveys*, 55(6), 1-28. <https://doi.org/10.1145/3530811>

- [26] Seyed Alinezhad, H., Roohi, M. H., & Chen, T. (2022). A review of alarm root cause analysis in process industries: Common methods, recent research status and challenges. *Chemical Engineering Research and Design*, 188, 846-860. <https://doi.org/10.1016/j.cherd.2022.10.041>
- [27] Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. da P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137, 106024. <https://doi.org/10.1016/j.cie.2019.106024>
- [28] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., et al. (2020). Explainable artificial intelligence: Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82-115. <https://doi.org/10.1016/j.inffus.2019.12.012>
- [29] Wickens, C. D., Li, H., Santamaria, A., Sebok, A., & Sarter, N. B. (2020). Stages and levels of automation: An integrated meta-analysis. *Human Factors*, 62(1), 4-22. <https://doi.org/10.1177/0018720818811140>
- [30] Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., et al. (2021). Knowledge graphs. *ACM Computing Surveys*, 54(4), Article 71. <https://doi.org/10.1145/3447772>