Intelligent Alarm Flood Management for Power Systems

Georgios Mitrentsis*, Susanne Schmitt*, David Marino‡ ,
Jhelum Chakravorty‡, Giancarlo Dalle Ave‡, Faeza Hafiz†, Antony Hilliard‡
Hitachi Energy Research
*Mannheim, Germany
‡Montreal, Canada
†Raleigh, North Carolina, USA

***Abstract*—Alarm floods pose a challenge to the successful operation of transmission and distribution systems because the alarm rate is often higher than what operators can effectively manage. Such a high volume of alarms can distract operators from unrelated but relevant alarms that occurred during the flood. To mitigate this situation, we propose a method to group alarm floods into clusters and identify additional alarms that might be lost in the flood and ignored by the operator. To do so, three different clustering approaches are introduced and tested using historical alarm data from a transmission system operator (TSO). Importantly, the results are validated by the end user highlighting the applicability of the proposed alarm flood management tool in real world scenarios.**

***Index Terms*—Alarm flood, control room, pattern mining, power system operation, unsupervised learning**

I. Introduction

Alarms play an important supportive role in preventing, detecting, controlling, and mitigating the effects of abnormal situations that require operator attention. Nevertheless, an inefficient alarm system can distract operators from important information and increase operator workload. For instance, the transmission system operator (TSO) of Croatia shared that they experienced over 100 alerts within 10-minute intervals during a single month [1]. Meanwhile, the Portuguese distribution system operator (DSO) E-REDES reported an average of 295,000 alarm occurrences per day, far exceeding the cognitive limits of what humans can reasonably respond to [2].

The situation becomes worse when alarm floods occur. In an alarm flood condition, the alarm rate is greater than an operator can manage (e.g., more than 10 alarms per 10 minutes). This causes a decline in situational awareness that might jeopardize the normal operation of the various grid components or even the stability of the power system. Hence, intelligent alarm management is of utmost importance for ensuring the reliable grid operation [3].

In this context, [4] is one of the first studies discussing the importance of intelligent alarm processing in power systems, while [5] proposes an on-line fault diagnosis system based on alarm messages. Reference [6] introduces a temporal constraint network to identify which events led to the recorded alarms and spot alarms that failed to be activated. Similarly, [7] employs Mixed Integer Linear Programming (MILP) to determine incomplete or incorrect sensor alarms presented to the system operator. The work of [2] introduces two innovative data-driven applications using Supervisory Control and Data Acquisition (SCADA) data: one for detecting unusual performance in protection relays during circuit breaker alarms, and the other for unsupervised clustering of similar events in high voltage line panels and classifying new event logs based on these clusters, including the identification of rare events. Another two data-driven approaches to support operators were introduced in [8]. The first one classifies the complexity of short-circuit faults based on alarm events, and the second one provides rapid guidance to operators on how to resolve these faults by suggesting suitable sequences of switching actions. However, these inspiring works focus on specific power system faults/events, such as short-circuits and protection relays, while a system operator may additionally receive communication, market-related, or other less severe power system alarms, e.g., voltage violations, temperature indications in power lines, etc. In addition, alarm floods in power systems seem to not have received notable attention in the existing literature yet.

Intelligent alarm flood management is a rather unexplored topic in power systems literature compared to its well-established counterpart in the process industries. While the topic of alarm flood management has been extensively studied and refined within the context of process industries [9]–[17], the existing methods and approaches developed for these sectors cannot be readily applied to power systems. The fundamental reason behind this limitation lies in the stark disparity of dynamics between the two domains. Power systems differ from industrial plants in significant ways: they are more complex, operate significantly faster, and are considerably larger. These differences render the direct transfer of methods questionable.

In a focus group session with members of a TSO, we discovered that a major desire of operators was to have the ability to group alarms or events that are linked together, which helps organize the alarm feed, reducing the risk of alarm floods

The authors gratefully acknowledge funding by the German Federal Ministry of Education and Research (BMBF) within the Kopernikus Project ENSURE 'New ENergy grid StructURes for the German Energiewende'

obscuring relevant items or accidental acknowledgements. To this end, we introduce a two-stage data-driven method to help operators tackle alarm flood situations. To accomplish this, the proposed alarm flood management scheme compresses alarm floods, identifies common alarm flood patterns among them, and detects additional alarms that might be lost in the flood and ignored by the operator. Those additional alarms are not generated by the same cause that trigger the alarm flood but rather, it happens to be raised at the same time period.

In the first stage, historical alarm data generated by a power system control software are analyzed and the alarm floods are identified. Next, each alarm flood is assigned to a cluster with similar alarm flood properties (e.g., protection events before the opening/closing of a circuit breaker, energy market related alarms, etc.). To do so, three different approaches are introduced, and four clustering algorithms of different notion and complexity are tested. Once the flood clusters have been formed, a general alarm flood pattern is extracted for each cluster using a pattern mining technique.

In the second stage, additional alarms that are not part of the flood, but they occur within the same time period, are automatically detected and presented to the operator. For this purpose, an alarm flood is first assigned to one of the flood clusters derived in the first stage and then, it is compared to the respective alarm flood pattern. The alarms of the flood that are not part of the extracted cluster pattern are classified as alarms that do not belong to the flood and are presented separately to the operator. The rest of the alarms (part of the flood) are compressed from the alarm list and are presented as an individual entry. To evaluate the proposed methodology, real alarm data generated within a period of 8 months in the control room of a TSO are employed.

In summation, we contribute:

1) A first application of alarm flood clustering techniques in the power system industry. Grouping similar alarm floods in power systems is a uniquely rather challenging task due to some inherent flood characteristics discussed later in Section III-A.
2) We propose a method to increase the situational awareness of the operator and thus, the reliability of the power system, by identifying alarms that may be lost in the flood due to the high volume of alarms happening at the same time.
3) This work employs the full raw operational dataset of control room alarms (power system, market, communication, test alarms) that the operator sees during the daily operation and not a subset of it focusing on specific types of faults.
4) We introduce a full data-driven approach that does not require any information about the power system model or topology.
5) Contrary to most of the works in the field of alarm management, the proposed methodology was validated by the end user (TSO).

II. Data Description and Preprocessing

Alarms floods are time intervals during which alarms are raised at a high rate so that the operators may have difficulties reacting to all of them in the right way. For the process industry, there is a standard for alarm flood threshold which is at least 10 alarms within 10 minutes, see e.g., [18]. However, we realized with the dataset at hand that by applying these thresholds, the system would be in flood state most of the time. In order to extract alarm floods which are reasonable and separated in time, we applied a stricter threshold of at least 20 alarms per minute and a total length of at least 10 alarms. These parameters were chosen by manual tests; a constructive method for determining such thresholds to facilitate flood clustering would be subject to future work. Thus, all alarms in time intervals that exceed these thresholds are extracted and stored as alarm floods.

For this work, the event list of a TSO is used, which consists of roughly 8.6 millions events, covering 8 consecutive months. Note that not all those events are alarms. An event can be any entry written in the SCADA database, while an alarm is what the operator receives in the control room. From this event list, a sublist of alarms was extracted based on their priority and information provided by the TSO. As a result, a list of 1.5 million alarms was extracted.

The alarms have multiple fields containing different types of information. For this work, the following fields are relevant:

- `timestamp`: the time at which the alarm is raised
- `text`: text description of the alarm
- `station`: name of the station, i.e., power system substation from which the alarm is raised
- `SCADA object ID`: the alarm identifier which is connected a to SCADA object e.g., sensor/device associated with the alarm

Table I summarizes some properties of the extracted alarms floods. It shows that a significant number of floods consist only of alarms from one single SCADA object ID. The flood lengths both in number of alarms and time duration show a wide range.

For this work, the alarms floods consisting of only one SCADA object ID are excluded, since clustering, pattern, and outlier detection would not lead to any significant insights for them. Instead, they may point to badly configured alarms that should be fixed in order to improve the overall alarm system performance.

TABLE I: Statistics about extracted alarm floods: number of floods, their number of alarms and time duration. Mid: all floods; right: without floods with only one SCADA object ID.

	all floods	w/o single IDs
# floods	5827	4526
min # alarms in flood	11	11
median # alarms in flood	18	33
max # alarms in flood	1896	1896
min flood duration in s	0.2	0.2
median flood duration in s	2	1
max flood duration in s	208	208

III. Proposed Method

The new alarm flood management method for power systems consists of multiple steps as shown in Fig. 1. First, the alarm floods are extracted based on the thresholds described above and then, they are grouped based on one of the three proposed approaches (Section III-A). As a next step, a flood pattern is extracted for each cluster using alarm set operations (Section III-B). As a final step, each flood is compared to the respective cluster pattern and the flood alarms that are not included in the pattern are classified as outliers (do not belong to the flood).

A. Alarm Flood Clustering

This is the most challenging step as each alarm flood may highly vary on the total number alarms included and the lack of specific alarm sequence in similar alarm floods. The latter originates from the fact that the power grid is characterized by fast system dynamics and the recorded alarm timestamp corresponds to the time instance that the alarm was written in the SCADA database and not to the time that the alarm was raised by the remote terminal unit (RTU).

After the alarm floods have been extracted as described in Section II, a feature matrix X is computed, where the rows correspond to the floods and the columns to the respective features.

Three different types of features are proposed:

1) TF-IDF (Term Frequency-Inverse Document Frequency): In this method, we calculate the TF-IDF scores using all SCADA object IDs as vocabulary and the floods as documents [19]:
   For each SCADA object ID id:

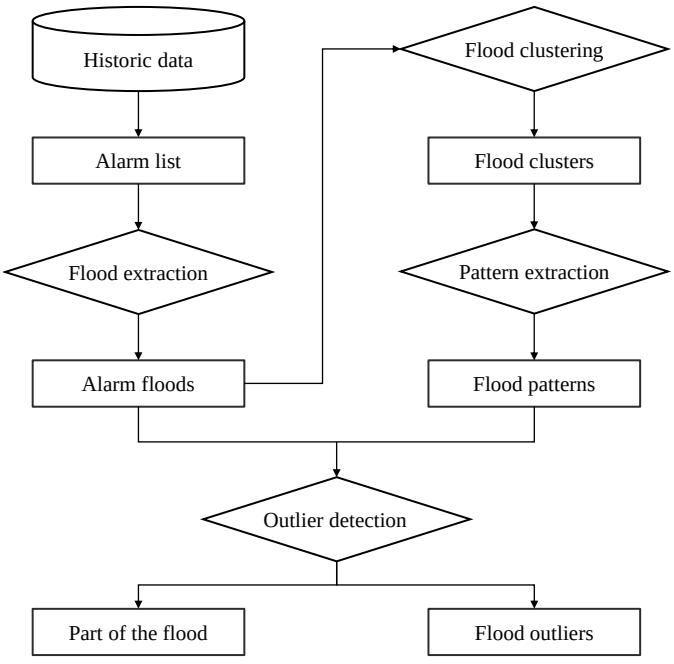

Fig. 1: Flow diagram of the overall method.

$$X_{flood,id} = \text{TF}(flood, id) \cdot \text{IDF}(flood, id), \quad (1)$$

where

$$\text{TF}(flood, id) = \frac{\text{count of } id \text{ in } flood}{\text{number of alarms in } flood} \quad (2)$$

is the document frequency of the SCADA object ID id in the alarm flood $flood$ and

$$\text{IDF}(flood, id) = \log\left(\frac{M}{\text{DF}(id)}\right), \quad (3)$$

where M is the total number of alarm floods and $\text{DF}(id)$ is the occurrence of id in all floods. To compute the TF-IDF scores, we used the Python package scikit-learn [20].

2) Occurrence matrix: In this case, the feature matrix X is compiled by binary entries as:

$$X_{flood,id} = \begin{cases} 1, & \text{if } id \in flood \\ 0, & \text{if } id \notin flood \end{cases} \quad (4)$$

In this approach, the feature matrix X has so many columns as the total number of alarms included in the floods.

3) Language model: In this approach the feature matrix contains sentence embeddings of the floods' SCADA object IDs as:

$$X_{flood} = encode(\text{“}id_1\ id_2\ ...\ id_{N_{flood}}\text{”}), \quad (5)$$

where the $encode$ function can represent any pre-trained sentence transformer and N_{flood} denotes the index of the last alarm in the flood sequence. Sentence embeddings have been developed to derive a numeric representation of semantic information in natural language processing. Embeddings are basically an encoding of a natural language sentence to numerical values. As similar natural language sentences produce identical embeddings, similar alarm floods expressed in text sequences of SCADA object IDs might yield also similar embeddings. It is worth mentioning that the SCADA object IDs are not natural language. However, the sentence embeddings can help to derive common combinations which appear in the floods and thus, enable clustering. For this approach, we use the recent BERT (Bidirectional Encoder Representations from Transformers) model [21]. Note that this is a pre-trained language model and no hyperparameter tuning is required.

Based on the feature matrix X, the alarm floods are then clustered using a clustering algorithm. For this, we have implemented the options of k-means, agglomerative hierarchical, spectral clustering and DBSCAN. For all the clustering algorithms, we use the Python package scikit-learn [20]. As a result of each clustering algorithm, a set of $\mathcal{C}$ flood clusters is obtained, each of them containing a set of the previously extracted alarm floods. Note that each alarm flood is mapped to exactly one flood cluster.

B. Pattern Detection

For each alarm flood cluster, a common pattern is determined. Initially, a set of the included SCADA object IDs is created for each alarm flood as $\mathcal{F}_j = \{id_1, id_2, ..., id_{N_j}\}$, where j denote the flood index and N_j is the total number of unique alarms included in the j-th flood. Next, the intersection of the flood sets belonging to each cluster is calculated as:

$$\mathcal{D}_c = \mathcal{F}_1 \cap ... \mathcal{F}_j \cap ... \cap \mathcal{F}_{M_c}, \tag{6}$$

where c is the cluster index and M_c is the total number of floods included in the c-th cluster. As a next step, the flood set with the lowest cardinality, i.e., minimum number of unique alarms, is found as:

$$\mathcal{S}_c = \min\left(|\mathcal{F}_1|, ..., |\mathcal{F}_{M_c}|\right). \tag{7}$$

Finally, the flood pattern of a cluster c is determined by the union of the two derived sets as:

$$\mathcal{P}_c = \mathcal{D}_c \cup \mathcal{S}_c. \tag{8}$$

This combination of sets has been empirically proved to be yielding the most representative pattern results, since it contains all the common alarms that are present in the floods and a few more that might be present in most of the floods.

It has to be emphasized that the pattern is considered as a set, i.e. the sequence does not matter. This is due to the fact that power systems alarms can be raised at very high frequency and the timestamps cannot be reliably associated with propagation of events through the system.

C. Outlier Detection

In this step, alarms that are contained in the floods but do not appear in the respective flood cluster pattern are detected. It is likely that those alarms do not belong to a flood, yet they are raised during the same time period rendering their detection by the operator challenging. These alarms should be individually displayed to the operator, whereas the rest of the flood alarms can be suppressed and displayed to the operator as a single entry.

To be precise, an alarm is classified as an outlier if it is not included in the flood pattern. This can be mathematically written as:

$$\mathcal{O}_j = \mathcal{F}_j \setminus \mathcal{P}_c, \tag{9}$$

where $\mathcal{O}_j$ represents a set of outliers for the j-th flood which has been assigned to the c-th flood cluster.

The outlier detection results are further refined by excluding the alarms that were raised in stations already included in the flood pattern. This is based on the assumption that if an alarm is raised in a station where a flood occurs, it is very likely that this alarm is also part of the flood.

TABLE II: Silhouette index for different clustering approaches and algorithms.

	Occurrence matrix	TF-IDF	Language model
k-means	0.456	0.455	0.343
Hierarchical	0.381	0.392	0.347
Spectral	0.134	0.436	0.363
DBSCAN	0.428	**0.494**	0.363

IV. Results

A. Alarm Flood Clustering Results

To evaluate the performance of the three proposed alarm flood clustering approaches, i.e., TF-IDF, occurrence matrix, and language model, we employ the average Silhouette index [22]. Silhouette index may range from -1 (worse) to 1 (best), while values close to 0 suggest clusters that overlap. Negative values imply that a sample has been erroneously assigned to a cluster whereas positive values indicate the opposite. Table II shows how the three proposed approaches performed using four different clustering algorithms of different notion and complexity. Overall, TF-IDF yielded the best clustering results regardless of the clustering algorithm. Occurrence matrix seems to be also a reliable approach, if it is combined with k-means or DBSCAN, whereas the deployed language model generated moderately worse yet consistent results across all algorithms. This decay in performance can be attributed to the fact that language models are usually trained using natural language and not coded tags like SCADA object IDs. Nevertheless, this language model could generate acceptable clustering results as validated by the authors by observing the individual flood clusters and the alarms therein. As for the total number of clusters, the well-established elbow method was employed [23].

It is worth pointing out the alarm flood clustering can be deployed as a standalone application, as also indicated in [11], [12], [16], [17], [24], [25]. Alarm engineers, managers, operators, and data analysts can leverage alarm flood clusters

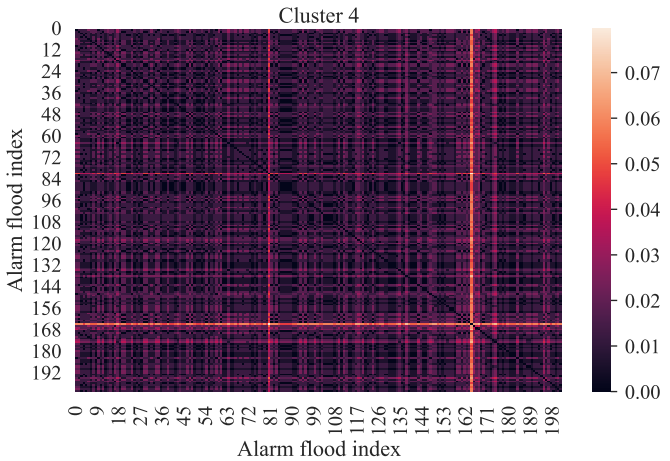

Fig. 2: Cluster 4: Heatmap of the pairwise Jaccard distances between the floods.

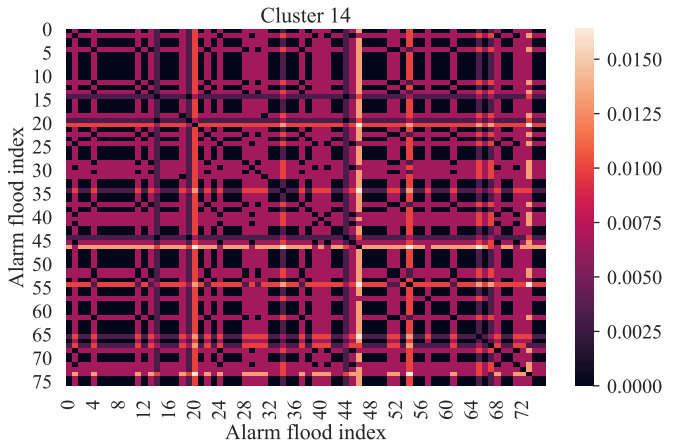

Fig. 3: Cluster 14: Heatmap of the pairwise Jaccard distances between the floods.

to identify error-prone equipment in the system, extract patterns for alarm suppression, perform root cause analysis, and support operators on their daily work [16].

In this context, Fig. 2 and 3 show the distance matrices of two indicative clusters generated using the TF-IDF approach and the DBSCAN algorithm. Each pixel corresponds to the Jaccard distance J_D between the different pairs of floods $(\mathcal{F}_i, \mathcal{F}_j)$, which can be calculated as:

$$J_D(\mathcal{F}_i, \mathcal{F}_j) = 1 - \frac{\mathcal{F}_i \cap \mathcal{F}_j}{\mathcal{F}_i \cup \mathcal{F}_j}. \tag{10}$$

If two floods are identical (their intersection and union are similar), then the second term of (10) will be close to 1 and thus, the Jaccard distance will be close to 0. If two floods have no alarm in common, then the Jaccard distance will be 1. In Fig. 2 and 3, darker pixels imply high similarity (low Jaccard distance) between the alarm flood patterns. As illustrated in those figures, the clustering results exhibit remarkable uniformity (Jaccard distance lower than 0.07), reflecting a high degree of accuracy in the grouping of the various alarm floods. This uniformity within each cluster further supports the efficacy of the clustering, as it displays consistent similarity patterns among cluster members.

B. Pattern and Outlier Detection Results

Once the flood clusters have been formed, a general flood pattern is determined for each cluster as described in the previous section. Based on this pattern, the alarms that do not belong to the flood (outliers), but they occured during the same time period, are identified and are presented separately to the operator. Since there is no ground truth to quantify the outlier detection rate, we developed a binary grid for each alarm flood cluster in order to visually inspect the output results. Due to confidentiality reasons, each alarm is represented by its index.

Fig. 4 is an representative example of the aforementioned binary grid of Cluster 8. Due to lack of space, we present only the first 20 alarm floods. Each alarm flood corresponds to a different horizontal line whereas each vertical line corresponds to a different alarm. The alarms are sorted based on their total contribution to the floods found in the cluster. For instance, the alarms that appear almost in every flood are placed on the left side of the grid, whereas the most rare ones can be found on the right side. This sorting enables the identification of alarms that do not belong to the flood and it is used to validate the outlier detection results as yielded by the proposed method.

In this regard, the alarms marked by the green rectangle indicate the alarm flood pattern since they appear almost in each flood or they were raised in the same substation as the other alarms of the pattern. On the contrary, the alarms marked by the red rectangles appeared only once and thus, it is very likely that they are not part of a flood. It is worth pointing out that a false classification of an alarm as an outlier would not have any severe consequence to the power system operation since outliers are presented separately to the operators. On the contrary, the false classification of an outlier as part of the flood might have the opposite effect. Therefore, operators should be cautious when suppressing alarm floods.

V. User Study

A user study was conducted to understand the qualitative impact of our alarm flood management tool. We recruited four expert users from a TSO. The participants were recruited using an inter-company agreement to share resources to develop experimental alarm management algorithms. Their roles along with their participant IDs are summarized in Table III.

We showed users a low fidelity excel based prototype that displayed the outputs of our clustering algorithm, and then conducted an outlier membership assessment activity, followed by a semi-structured interview [26]. The excel based prototype was created by running our algorithm on the user's own alarm system data. A single sheet was an alarm cluster. The sheet displayed alarm data in two columns: one column showed the common pattern shared by all alarms, the second column showed outliers that may or may not be a part of the flood. We showed users three flood clusters during each session. Flood clusters were selected using the following principles: Cluster 1 (C1) had a balanced number of outliers and common patterns, Cluster 2 (C2) had more common patterns than outliers, and Cluster 3 (C3) had more outliers than common patterns. To select a cluster, a single researcher randomly sampled flood clusters until it met one of the criteria for C1, C2, or C3. We conclude our user study by summarizing our interviews and

TABLE III: Expert users and their roles recruited in our user study.

Participant ID	Role
P1	Head of System Data Operations Team
P2	Market Operator
P3	IT Manager
P4	Realtime Systems Engineer

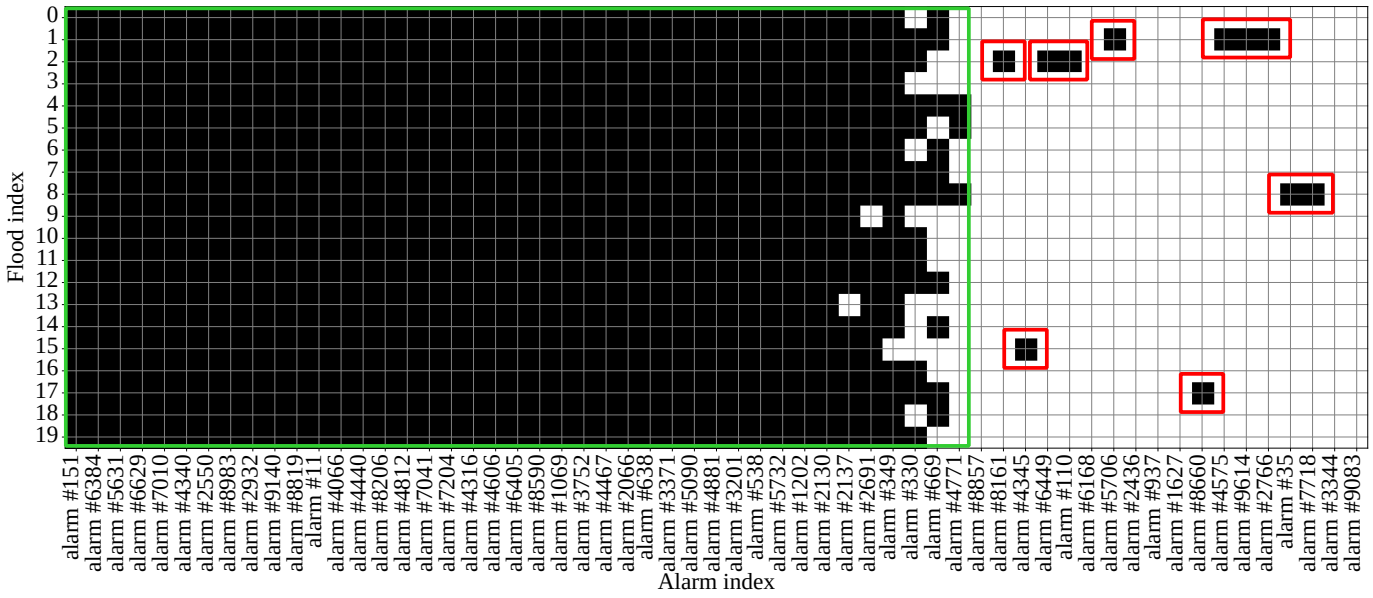

Fig. 4: Cluster 8: Binary grid illustrating the alarms that belong or do not belong to an alarm flood. Black pixels denote that an alarm is present in a flood whereas white pixels indicate the opposite. Each horizontal line corresponds to a different flood while the each vertical line corresponds to a specific alarm.

relating them to our research questions. Our research questions were specifically as follows:

1) **RQ1 - Cluster Meaning**: How meaningful are the clusters to end users in being able to understand the state of the power system?
2) **RQ2 - Outlier Accuracy**: How should the relationship between outliers and common patterns be presented to end users?
3) **RQ3 - Expert Improvement**: How can future versions of the flood clustering algorithm be improved?

A. Outlier Membership Assessment

We asked participants to sort outliers for all alarms based off whether they thought they truly belonged to the alarm flood, or if they were erroneously classified. Participants were given the option of saying "belongs", "does not belong", or "do not know". In total, participants sorted 26% of outliers as definitely belonging to the flood cluster. P4 said that given more context, he could have seen additional alarms as possibly belonging to the flood. With these alarms included, 37% would have then been perceived as belonging. Participants further ranked 29% of alarms as not belonging to the flood. The remaining 45%-34% was unknown.

B. User Interview

Here our interview questions are iterated and responses summarized. We link to our research questions to interview responses the following section.

1) ***Q1*** *Do you have an intuition as to the cause of this flood?:* The goal of our evaluation was not to determine if their rationalizations were correct, but rather if the floods were sufficiently meaningful to users to rationalize about potential flood causes. P2 and P4 were able to rationalize about the presented alarm flood clusters, and determine possible causes of floods. For instance, they indicated topology change and DC cable switching as possible causes of some floods. In contrast, P1 and P3 were not able to rationalize about the flood clusters and chose to abstain from answering the question. The distinction between these two user groups was that P2 and P4 had direct operations experience, where P1 and P3 were both managers and more removed from daily alarm list activities. This implies that the clusters have more utility to operators than managers.

2) ***Q2*** *Can you imagine a time when you would do the same action on all of these alarms? (e.g. block, acknowledge, etc.):* P1 and P2 recognized that the alarms we presented were the responsibility of a specific organizational unit within the TSO, and identified a particular control center to call with the presented clusters. P4, whose responsibilities were directly relevant with the presented alarms, noted that often times the causes of a flood can be complex, and that an operator would need to investigate the substations and connective neighbourhood to truly understand which actions to take.

3) ***Q3*** *When it comes to flood clusters such as these, could you see yourself using them in your day to day activities?:* P4 responded positively to the flood clustering functionality, particularly about the promise in it reducing the length of the alarm list:

> "Yes...we receive a lot of alarms and if you can combine [them] into one item, that gives a better overview of what's going on and then you don't miss other important items...if you have a long list that then you are a little bit lost. " –P4

P2 required more transparent information about what principles were used to create alarm clusters (i.e. distance measures) before they were comfortable adopting it in their everyday practice. The other two participants abstained from commenting.

*4) **Q4** What are some principles you may find useful to group alarms?:* All participants mentioned geographic location as a major principle to group alarms. P4 further offered geographic location as a heuristic to sort outliers as belonging or not belonging. Some more grouping heuristics discussed were: alarm type, priority, *a priori* groups, and electric connectivity.

C. Relating User Findings to Research Questions

We understand the interviews as answering **RQ1 - CLUSTER MEANING** as follows: the clusters are most meaningful to operators and those that need to handle alarms on an every day basis. They provide a sufficient platform to hypothesize on possible root causes of floods, but do not tell the whole story. Further investigation is required by operators to ultimately determine what actions to take; the main perceived benefit of the clusters is in helping operators organize their alarm list view as to manage cognitive load.

We answer **RQ2 - OUTLIER ACCURACY** through the findings of our Outlier Membership Assessment Task (Section V-A). Outliers were perceived as being uniformly split between being true flood members ($26\% - 37\%$), not members (29%), or unknown ($45\% - 34\%$). When displaying the outputs of our flood clustering technique, it is therefore important to also represent the outliers, and not hide them. The full outlier list should be shown so that the users can determine belonging on their own. The outlier list should also incorporate visual design choices that demonstrate uncertainty, such that there is a 1/3rd chance that an item in the list is a true member of the alarm flood.

We relate interview responses to **RQ3 - EXPERT IMPROVEMENT** in the following ways: location information can further be incorporated into the system to aid operators since much attention when rationalizing groups was spent on geographic location. Using geographic labels for clusters may be an intuitive way to describe clusters to end users since they often describe floods in location-based terms themselves.

VI. CONCLUSION AND FUTURE WORK

Traditional flood management approaches are likely to fail in the power system context since they are faster, span vast geographical areas, and have more complex dynamics. To this end, we introduced a method that supports the operator in processing high volumes of alarms.

We evaluated our algorithm using real alarm data obtained from a period of 8 months from a TSO control room. The proposed approach was able to distinguish different flood patterns and group similar alarms floods in the same cluster. Specifically, we demonstrated a uniform pairwise similarity between alarm floods using the Jaccard distance metric, and assessed overall clustering performance using the Silhouette coefficient. The TF-IDF approach showed consistent clustering performance and its combination with the DBSCAN produced the best clustering results.

Furthermore, our user study discovered that our flood clusters had utility in giving operators a basis to hypothesize about potential flood causes, and showed the most promise as a tool to better organizing the alarm feed, thereby reducing cognitive load. As for the outlier detection, the proposed method was able to identify some alarms that were indeed not part of the floods leading to an increase in situational awareness, as the risk of missing important alarms that may be lost in the flood is decreased. Nevertheless, robust outlier detection would ideally need operators annotating the dataset.

Future work will focus on incorporating the outcome of the user study and integrate it into the development of the proposed alarm flood management tool.

REFERENCES

[1] N. Baranovic, P. Andersson, I. Ivankovic, K. Zubrinic-Kostovic, D. Peharda, and J. E. Larsson, "Experiences from intelligent alarm processing and decision support tools in smart grid transmission control centers," in *Cigre Session*, vol. 46, 2016, pp. 21–26.

[2] J. R. Andrade, C. Rocha, R. Silva, J. Viana, R. J. Bessa, C. Gouveia, B. Almeida, R. Santos, M. Louro, P. Santos *et al.*, "Data-driven anomaly detection and event log profiling of SCADA alarms," *IEEE Access*, vol. 10, pp. 73 758–73 773, 2022.

[3] L. Wei, W. Guo, F. Wen, G. Ledwich, Z. Liao, and J. Xin, "An online intelligent alarm-processing system for digital substations," *IEEE transactions on Power Delivery*, vol. 26, no. 3, pp. 1615–1624, 2011.

[4] D. S. Kirschen and B. F. Wollenberg, "Intelligent alarm processing in power systems," *Proceedings of the IEEE*, vol. 80, no. 5, pp. 663–672, 1992.

[5] H. Miao, M. Sforna, and C.-C. Liu, "A new logic-based alarm analyzer for on-line operational environment," *IEEE Transactions on Power Systems*, vol. 11, no. 3, pp. 1600–1606, 1996.

[6] W. Guo, F. Wen, Z. Liao, L. Wei, and J. Xin, "An analytic model-based approach for power system alarm processing employing temporal constraint network," *IEEE Transactions on Power Delivery*, vol. 25, no. 4, pp. 2435–2447, 2009.

[7] Y. Jiang and A. K. Srivastava, "Data-driven event diagnosis in transmission systems with incomplete and conflicting alarms given sensor malfunctions," *IEEE Transactions on Power Delivery*, vol. 35, no. 1, pp. 214–225, 2019.

[8] V. Campos, J. R. Andrad, R. J. Bessa, and C. Gouveia, "ML-assistant for human operators to solve faults and classify events complexity in electrical grids," in *13th Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MEDPOWER 2022)*, vol. 2022, 2022, pp. 336–341.

[9] B. Zhou, W. Hu, K. Brown, and T. Chen, "Generalized pattern matching of industrial alarm flood sequences via word processing and sequence alignment," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 10, pp. 10 171–10 179, 2020.

[10] M. Lucke, M. Chioua, C. Grimholt, M. Hollender, and N. F. Thornhill, "Advances in alarm data analysis with a practical application to online alarm flood classification," *Journal of Process Control*, vol. 79, pp. 56–71, 2019.

[11] I. Weiß, J. Kinghorst, T. Kröger, M. F. Pirehgalin, and B. Vogel-Heuser, "Alarm flood analysis by hierarchical clustering of the probabilistic dependency between alarms," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. IEEE, 2018, pp. 227–232.

[12] V. Rodrigo, M. Chioua, T. Hagglund, and M. Hollender, "Causal analysis for alarm flood reduction," *IFAC-PapersOnLine*, vol. 49, no. 7, pp. 723–728, 2016.

[13] M. R. Parvez, W. Hu, and T. Chen, "Comparison of the Smith-Waterman and Needleman-Wunsch algorithms for online similarity analysis of industrial alarm floods," in *2020 IEEE Electric Power and Energy Conference (EPEC)*. IEEE, 2020, pp. 1–6.

[14] M. Lucke, M. Chioua, C. Grimholt, M. Hollender, and N. F. Thornhill, "Online alarm flood classification using alarm coactivations," *IFAC-PapersOnLine*, vol. 51, no. 18, pp. 345–350, 2018.
[15] T. Niyazmand and I. Izadi, "Pattern mining in alarm flood sequences using a modified PrefixSpan algorithm," *ISA Transactions*, vol. 90, pp. 287–293, 2019.
[16] B. Zhou, W. Hu, and T. Chen, "Pattern extraction from industrial alarm flood sequences by a modified CloFAST algorithm," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 288–296, 2021.
[17] K. Ahmed, I. Izadi, T. Chen, D. Joe, and T. Burton, "Similarity analysis of industrial alarm flood data," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 452–457, 2013.
[18] "Alarm systems; guide to design, management, and procurement," Engineering Equipment and Materials Users Association, London, UK, Standard, 2015.
[19] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.
[20] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.
[21] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: http://arxiv.org/abs/1908.10084
[22] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
[23] M. Syakur, B. K. Khotimah, E. Rochman, and B. D. Satoto, "Integration k-means clustering method and elbow method for identification of the best customer profile cluster," in *IOP conference series: materials science and engineering*, vol. 336. IOP Publishing, 2018, p. 012017.
[24] G. Manca, M. Dix, and A. Fay, "Clustering of similar historical alarm subsequences in industrial control systems using alarm series and characteristic coactivations," *IEEE Access*, vol. 9, pp. 154 965–154 974, 2021.
[25] S. Lai and T. Chen, "A method for pattern mining in multiple alarm flood sequences," *Chemical Engineering Research and Design*, vol. 117, pp. 831–839, 2017.
[26] W. C. Adams, "Conducting semi-structured interviews," *Handbook of practical program evaluation*, pp. 492–505, 2015.