17th International Conference on Probabilistic Safety Assessment and Management &
Asian Symposium on Risk Assessment and Management (PSAM17&ASRAM2024)
7-11 October, 2024, Sendai International Center, Sendai, Miyagi, Japan

Towards the Use of Temporal Convolutional Networks for Guiding Dynamic Risk Assessment Methods

Renan G. Maidana[a*], Yiliu Liu[b]
[a] Department of Energy and Transport, SINTEF Ocean
[b] Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology

Abstract: Risk assessment is a well-established process and is often used to assess the risk of systems in several application domains, e.g., maritime, nuclear power, and aerospace. However, the results from a risk assessment represent a snapshot of the risk picture for the system at a given time and, for dynamic systems whose states can change suddenly, these results become obsolete after a while. Dynamic risk assessment methods consider the ever-changing and uncertain nature of dynamic systems. A subset of these methods consists of simulating how the system states evolve in time while considering rules, events and actions that can lead to accident events with undesirable consequences. This position paper studies the use temporal convolutional networks to predict the probability of accident events given a time-series of events, to assist dynamic probabilistic risk assessment methods and improve their computational performance. The paper's contributions are a study of the properties of temporal networks and an evaluation of their feasibility for predicting the probability of future accident events in dynamic probabilistic risk assessment methods. Furthermore, an approach is proposed that uses a temporal network for probabilistic forecasting, i.e., predicting a probability distribution based on time series data. The output of this network could be used to inform and guide the dynamic risk assessment process.

Keywords: Dynamic risk assessment, machine learning, temporal convolutional networks, risk prediction

1. INTRODUCTION

Risk Assessment (RA) is the process of analyzing and evaluating risk [1], whose results can be used by decision-makers to ensure safe operation. New challenges emerge when performing RA of Autonomous Systems (AS), such as autonomous cars or Maritime Autonomous Surface Ships (MASS). These systems typically operate semi-autonomously (i.e., with a human supervisor) or with full autonomy (i.e., without direct human oversight) [2], and their risk picture changes compared to manned systems. For example, the group risk for the MASS crew members is reduced, as fewer people are present in a MASS. Simultaneously, the risk for passengers and crew of other ships increases, as AS are not as capable as humans in adapting to uncertain environments and previously unknown situations [1].

Autonomous systems such as MASS can also be defined as dynamic systems, i.e., systems whose states can change over time. The results of traditional RA of AS are useful for a limited time due to these systems' dynamic nature, i.e., the results become less accurate and more uncertain as their states change. Dynamic Risk Assessment (DRA) emerged as a response to this issue, where RA for a system is performed while considering its dynamic and uncertain nature. Typically, methods for DRA consist of evolving models of the dynamic system in time from an initial state, and evaluating whether certain events in the system's operation timeline will lead to outcomes with undesirable consequences (i.e., accident events) [3]. The sequences of events leading to accident events are “accident scenarios”, i.e., sequences of low-level events (e.g., engine failure) which cause deviations in operation, leading to high-level accident events such as, collision of a MASS with another vessel [3]. When a sequence of events on a system exposed to risk leads to desirable outcomes (e.g., avoiding a collision), the sequence is defined as a “risk scenario”.

The objective of DRA is to find all (or as many as possible) risk and accident scenarios for a dynamic system given an initial state. A common variant of DRA is Dynamic Probabilistic Risk Assessment (DPRA), where the probabilities of the scenario events and of the scenario itself are quantified. However, most DPRA methods suffer from a combinatorial explosion problem due to the high complexity of large-scale systems, leading to high computational requirements and low performance [3]. This problem is known as the state explosion problem, and solutions to it are known as supervised or guided DPRA. These solutions typically consist of reducing the search state-space [4, 5], for example, by pruning states with very low probability, or guiding the evolution of risk and accident scenarios towards the ones with highest risk [6] with heuristic functions or biases. A caveat to supervised DPRA is that although the state explosion problem is alleviated, the computational burden can still be high for large systems. Furthermore, there is a trade-off between completeness (i.e., finding all scenarios) and computational performance, and users must balance exploring the state-space of scenarios and finding the most important ones in feasible time (i.e., with enough time to act in an emergency).

Predicting the probability and severity of accident scenarios before simulating them would reduce the uncertainty of future events and simultaneously improve the computational efficiency of DPRA methods, as the most critical scenarios could be prioritized. Data-driven estimators from machine learning have been used to predict the probability of accident events in risk scenarios. For example, Patel and Liggesmeyer [7] used an artificial dataset built from a simulator to train support vector machine models to predict the probability of collision accidents for an autonomous car. Feth et al. [8] used a Convolutional Neural Network (CNN) to assess the risk of collision, where the CNN is trained with images of various traffic scenes from a stereo camera, also generated via simulation, and an associated risk metric. Wang and Kato [9] trained a CNN for assessing the collision risk, training the model with images from YouTube videos, annotated in three levels of risk. More recently, Tritsarolis et al. [10] proposed a framework with deep learning models to forecast the route of ships in collision encounters, given historical automatic identification system data, and determine their collision risk index.

Considering that the probability of accident scenarios changes over time in dynamic systems, depending on their behavior and operating context, their risks may be expressed as time series. Temporal Convolutional Networks (TCN), a subset of deep learning methods, specialize in identifying patterns and making predictions based on time-series data, with applications action classification in videos with Recurrent Neural Networks (RNN) or Long-Term-Short-Memory (LSTM) networks, for example. In this position paper, the properties of TCNs are studied to evaluate their feasibility towards DRA and, more specifically, DPRA problems. We discuss the concept of probabilistic forecasting and how risk and the generation of Discrete Dynamic Event Trees (DDET), a model commonly used for DPRA, relate to this concept. We then propose using deep TCN, a probabilistic forecasting framework, to find the probability distribution of future accident events in DDETs, possibly guiding the scenario generation process in DPRA. Note that, as a position paper, the concept of probabilistic forecasting and the approach for applying it to DDET generation are discussed and proposed only. Implementation and demonstration in case studies will follow as future works.

The rest of this paper is organized as follows. Section 1 briefly introduces DPRA and TCNs. Section 2 discusses probabilistic forecasting and presents the proposed approach for using TCNs to assist DPRA. Section 3 discusses the possible advantages, challenges and caveats of this approach, and Section 4 presents conclusions, limitations, and future works.

2. BACKGROUND

In this section, the areas of DPRA and TCNs are briefly presented. More specifically, we introduce DPRA and present a subset of methods for DPRA based on DDETs, the main idea and structure of TCNs, and the concept of probabilistic forecasting, which will be explored in Section 3.

2.1. Dynamic Probabilistic Risk Assessment

As discussed in Section 1, DPRA methods consist of evolving the state of a dynamic system in time to find accident scenarios that emerge from the interaction between events, e.g., a process variable out of scope, a component failure, or human error. Methods for DPRA are typically concerned with the evolution of risk scenarios from an initiating event, commonly achieved by building dynamic event trees, located on the right-hand side of the bow-tie diagram, as shown in Figure 1. On the other hand, dynamic risk models such as dynamic fault-trees [18, 19] and dynamic Bayesian belief networks (BBN) [13] consider the causal effect of hazards and threats on initiating events, located on the left-hand side of the diagram.

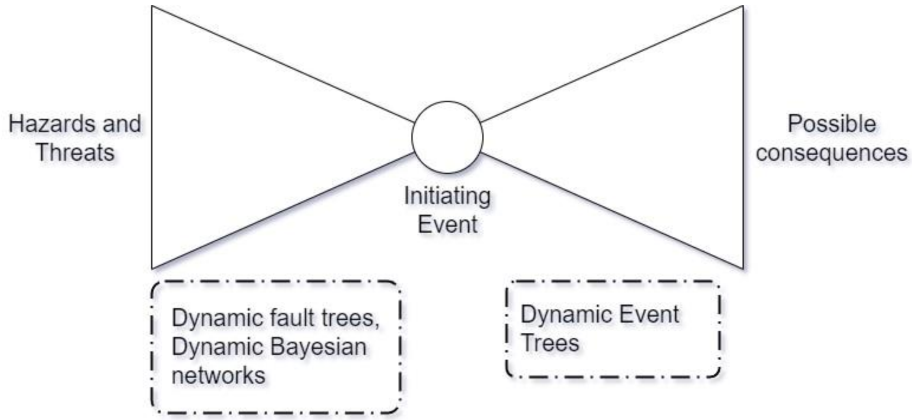

Figure 1. Bow-tie diagram illustrating the connection between causes (i.e., hazards and threats), an initiating event, and the possible consequences.

DETs can be represented in continuous-time as Continuous DETs (CDET), or in discrete-time as DDETs. CDETs are typically represented as sets of equations for a dynamic system [3], solved for example with Monte-Carlo simulation [21-23] to obtain the risk and accident scenarios. On the other hand, DDETs are represented as simulation trees or graphs that hold risk scenarios, generated by simulating a system in time while considering several external and internal factors. The Accident Dynamic Simulator (ADS) [17] and the Simulation-based Probabilistic Risk Assessment (SimPRA) [18] are examples of frameworks for DDET-based DPRA. They work by simulating the physical model of a dynamic system in time while considering the effects of human error with cognitive models until an outcome or possible consequence occurs. The outcomes and consequences are defined as “end states”, and can be desirable (e.g., collision avoided) or undesirable (e.g., collision event). In ADS, for example, the simulation proceeds until a low-level event occurs, such as a process variable out of scope or a human operator failure. At that point, the simulation timeline splits into two branches, one with the probability of that event occurring and another with the probability of it not occurring. This process repeats for each branch until end states are found, and the risk scenarios are the sequences of events from the initial state to the end state. As discussed in Section 1, when the end state is an undesirable consequence, the scenario becomes an accident scenario. An example of a DDET can be seen in Figure 2.

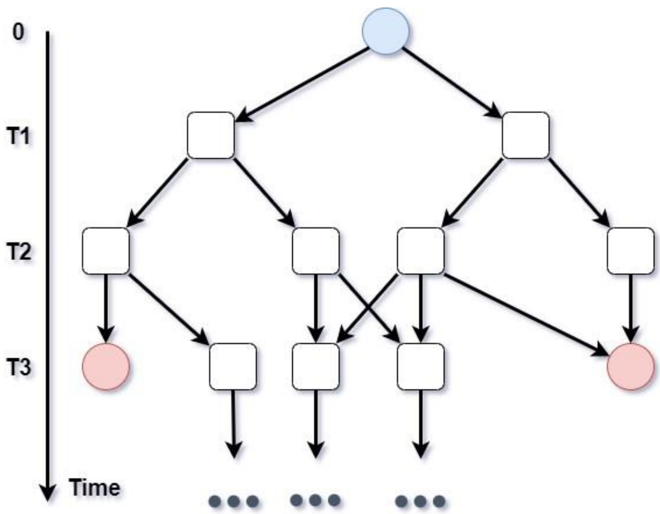

Figure 2. Example of a DDET. The blue circle represents the initial state, the squares are low-level events, and the red circles represent the end states that have been reached. The three dots indicate that the DDET continues downwards in some branches.

2.2. Temporal Convolutional Networks

In the area of deep learning, CNNs are typically used for pattern recognition and, more specifically, object classification in images [19]. They are suited to processing data represented in grid-like structures, such as arrays for 1-dimensional data (e.g., time series, sensor values), or matrices for 2-dimensional data (e.g., images). CNN models are built by performing mathematical operations (e.g., convolution, pooling) multiple times between the input data and a predicted output. An apt analogy is to imagine that the mathematical operations are building blocks, and CNN models are built by stacking different blocks between the input data and the predicted output. Weights at each block are adjusted using large, labeled datasets and optimization procedures, encoding knowledge about patterns in the input data and correlations between input and output. The process of "training" these networks with labeled data is known as "supervised learning" [19].

Although CNNs are suited to pattern recognition and classification in images, they are not well-suited to identify and classify patterns that depend on time, e.g., actions in videos. To illustrate, consider that a CNN's task is to evaluate a video of a child playing football on a field. The CNN can accurately classify each object in every frame of the video, i.e., it can classify the ball, the child's clothing, goal posts, and so on. However, it may struggle to identify the action of "playing football" because 1D and 2D CNNs are ill-suited to capture the temporal link between images. In other words, it may struggle to identify an action that occurs over several frames, contained in a time series of input data.

This problem, known as action recognition, is addressed by temporal networks. RNNs, for example, consider the causal links between past data and its prediction by letting the current prediction inform the next one recursively. In other words, a given frame depends on what has come before [19]. A popular approach to perform action recognition, called "dual-channel networks", handles space and time domain features separately [20]. More specifically, it consists of using a CNN to identify a space (e.g., the football field), for better understanding the context of an action, and a temporal network to identify the motion between frames (e.g., the child running to kick a ball). Two challenges in this approach are the definition of two separate learning models and that nuanced causal relationships may be lost between the CNN and the temporal network.

Temporal convolutional networks are "single-channel" models [21] proposed to address these challenges. It uses an encoder-decoder structure with temporal convolution layers that capture the temporal dependency between images and sensor data. These networks have been successfully used recently for predicting traffic trends using data from ride-sharing applications [22] and for probabilistic forecasting of decision-making in online businesses [23].

3. PROPOSED APPROACH

This section discusses the approach proposed for using temporal convolutional networks to assist in the DPRA process. More specifically, we discuss how to use the data from DDET-based DPRA to estimate the future probability of accident events, framed as a problem of probabilistic forecasting, which TCNs solve.

3.1. Probabilistic Forecasting

Temporal convolutional networks are used in [23] to estimate probability densities in time series data. More specifically, the authors propose a framework using a TCN in the autoencoder format proposed in [21], combined with representation learning[1], to identify time-dependent patterns and forecast the probability distribution of future time series. The framework, called deep TCN[2], is applied to five datasets, where it makes accurate predictions on probability densities of consumer shopping trends, electricity consumption, and market demand for spare car parts, for example. Furthermore, the framework can learn complex patterns from factors external to the datasets, such as seasonality and approaching holidays.

The problem solved by deep TCN is that of probabilistic forecasting, i.e., to estimate the conditional probability distribution of future time series by incorporating historical data [23]. Formally,

---

[1] A set of methods allowing for the automatic discovery of relevant representations or features in a dataset.
[2] https://github.com/oneday88/deepTCN

$$y_{1:t} = \{y_{1:t}^{i}\}_{i=1}^{N} \tag{1}$$

is a set of N time series for a given system, where t denotes the length of the historical data considered in the forecast, i.e., $y_{1:t}$ represents a set of N time series up to the present time t. Then,

$$y_{t+1:t+\Omega} = \{y_{t+1:t+\Omega}^{i}\}_{i=1}^{N} \tag{2}$$

represents the set of future time series data, where Ω is the length of the forecasting horizon. Probabilistic forecasting aims to predict the probability distribution P of the future time series, i.e., how likely it is that the future data will appear, formally, $P(y_{t+1:t+\Omega} \mid y_{1:t})$. The approach to computing P in deep TCN is a modified version of the conditional probability product used in generative models, which compute P as the product of conditional probabilities in historical data [23]. Generative models compute this probability in a dead-reckoning manner as the product of the pairwise conditional probabilities in historical data. In deep TCNs, the probability distribution is conditional on all the historical data:

$$\mathrm{P}(\mathrm{y}_{\mathrm{t+1:t+\omega}} \mid \mathrm{y}_{\mathrm{1:t}}) = \Pi_{\omega=1}^{\Omega} \mathrm{p}(\mathrm{y}_{\mathrm{t+\omega}} \mid \mathrm{y}_{\mathrm{1:t}}) \tag{3}$$

In other words, the data series y_{t+1} is conditional on all the historical data $y_{1:t}$, as is the data series y_{t+2} and so on until the forecast horizon is reached. Another addition to deep TCN's computation of P is the "covariates" X, representing sets of factors external to the data in the time series but which nevertheless affect their probabilities. An example of a covariate for market demand forecasting is whether a big holiday like Christmas is approaching. The formal definition of P in deep TCN with covariates is:

$$\mathrm{P}(\mathrm{y}_{\mathrm{t+1:t+\omega}} \mid \mathrm{y}_{\mathrm{1:t}}) = \Pi_{\omega=1}^{\Omega} \mathrm{p}\left(\mathrm{y}_{\mathrm{t+\omega}} \mid \mathrm{y}_{\mathrm{1:t}}, \mathrm{X}_{\mathrm{t+\omega}}^{\mathrm{i}}\right) \tag{4}$$

where $i = 1{:}N$. Deep TCN solves probability forecasting by predicting this probability distribution with temporal convolutional networks.

3.2. Temporal Convolutional Networks for DPRA

As discussed in Section 3.1, P represents a probability distribution for a set of future time series, where these probabilities are conditional on historical data, i.e., from somewhere in the past until the present time, as well as external factors. There are parallels between the problems of probabilistic forecasting and DPRA. As discussed in Section 2, DDETs can be built through simulation to perform DPRA in a dynamic system, for example, using frameworks such as ADS and SimPRA. Risk scenarios in a DDET evolve in time from an initiating event, which means an evolving scenario can be seen as a time series from $t = 0$ at the initiating event to $t = T$, where T denotes the present simulation time:

$$\mathrm{z}_{0:\mathrm{T}} = \{\mathrm{e}_{\mathrm{k}}\}_{\mathrm{k}=0}^{\mathrm{T}} \tag{5}$$

where e_k represents an event at time $t = k$ in the risk scenario. The objective of DDET-based DPRA is to find all possible risk scenarios given an initial state, with special interest in accident scenarios. The result of DDET-based DPRA is thus a DDET containing all possible scenarios. Therefore, the DDET can be seen as a set of evolving time series, i.e., scenarios:

$$y_{0:T} = \{z_{0:T}^{i}\}_{i=1}^{N} \tag{6}$$

where N is the number of possible scenarios in the DDET. As discussed in Section 1, generating DDETs incurs a high computational burden, as only a finite subset of a large state space of possible scenarios will have accident events. Considering that a DDET can be seen as a set of time series, an approach to assist the DPRA process would be to compute the probability distribution of future accident events with TCNs. More specifically, the probabilistic forecasting of future accident end states could be performed with deep TCN, for example, where the branching paths in a DDET are the historical data, defined as a set of time series as in Equation (6). The probability distribution predicted from the TCN can serve as a guide to generating the DDET, i.e., scenarios with a high probability of having an accident end state in the future could be marked for priority simulation. Furthermore, if desired, a probability threshold could be used to prune branches whose probability, given by the TCN, is below the threshold. In other words, branches with a probability below a certain threshold would be marked as "closed," meaning they would not be further simulated.

To illustrate, consider the DDET generation process depicted in Figure 3 to 5, where the DDET starts from an initial state (in blue) and grows downwards, with time increasing in discrete steps. For each simulation branch, probabilistic forecasting predicts the probabilities (in green) that the simulated scenarios (i.e., from the initial event to the current event) will result in an accident event in the future. In the beginning, the probability distribution is mostly uniform as there is not much historical data to draw from, as seen in Figure 3. However, after some time, the number of branches grows, meaning more historical data is available, and the probability distribution can better classify which scenarios may result in future accident events as seen in Figure 4. In other words, higher probabilities are predicted for scenarios that likely have future accident events according to the historical data. Finally, some end states are reached in Figure 5, which is reflected by the updated probability density. These scenarios are recorded (i.e., closed), and the probabilistic forecasting process continues for the remaining branches which have not yet reached end states.

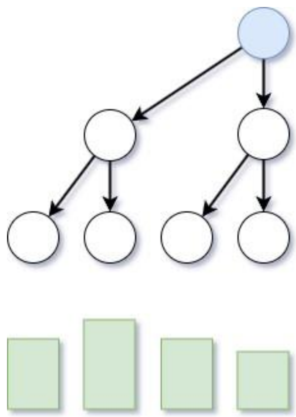

Figure 3. Probability distribution for a DDET after two time-steps. The probability distribution is mostly uniform due to the lack of historical data.

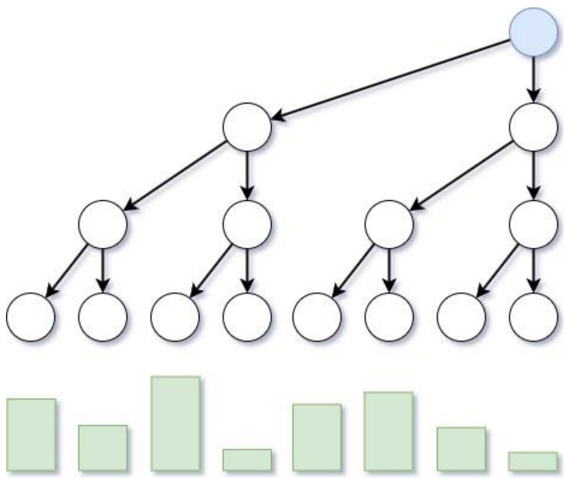

Figure 4. Probability distribution for a DDET after three time-steps. The probability distribution is refined with extra historical data.

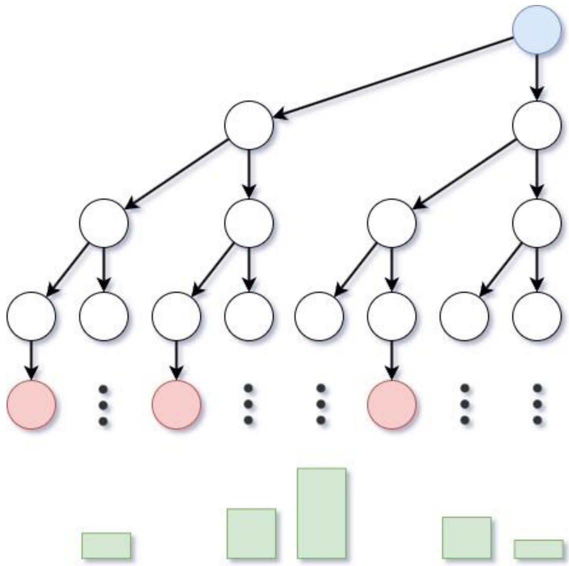

Figure 5. Probability distribution for a DDET after four time-steps. Some accident end states (red) are removed from the distribution after being found. The forecasting and simulation continue for the remaining open branches, represented with the three dots.

3. DISCUSSION

This section discusses the advantages, challenges, and caveats to using TCN to assist DDET-based DPRA.

3.1. Improving DDET-based DPRA with Temporal Convolutional Networks

In [24], the DDET-based DPRA problem was framed as a relaxed version of the K-Shortest-Paths (KSP) problem, named "relaxed KSP," with the objective of addressing DPRA's state-explosion problem using algorithms with low computational complexity. To that end, it was shown that DDET-based DPRA and relaxed KSP share the same requirements. Then, an approach called KPRA was proposed to generate DDETs with a heuristic search algorithm called K*, which solves the relaxed KSP problem. Finally, two ideas for improving K* were proposed, one of them involving machine learning methods to assist in finding the accident scenarios.

This idea depends on the assumption that, in an accident scenario, there is a correlation between the events preceding an undesired consequence and the accident event itself, i.e., the occurrence of an accident event depends on the events leading up to it. Thus, a binary predictor could be used whenever a new DDET branch is generated. The predictor would consider the events leading up to that branch to evaluate if an accident event is likely to happen down the line or not. In the latter case, that branch would be pruned from the simulation tree, reducing the computational load, and further improving the algorithm's performance. This idea implies that the events leading up to the evaluated branch are time series describing what happened in the past. As TCNs are specialized to identify patterns in time series data, the choice to investigate their use for DPRA was made, an intersection which has received little attention, as discussed in Section 1.

An advantage of using TCNs as proposed here is that they are supervised learning models which need to be trained with labeled datasets. Training a TCN to predict accident events means that the datasets used must represent accident scenarios in the real world. Therefore, the prediction does not depend, for example, on the manual definition of causal relationships, nor on the definition of a heuristic function that describes future risk, two tasks that are often non-trivial. In other words, the predictions are made based on patterns learned from the accident scenarios in the training sets, avoiding the need to define heuristic functions, for example. Furthermore, a TCN may be trained with a specialized dataset to identify a specific accident, such as ship collisions or capsizing. If specialized TCNs are trained for different accidents, an ensemble network model[3]

[3] A specialized machine learning model made through the union of several individual models [25].

may be constructed to predict the probabilities of end states considering different accident types simultaneously. Another promising idea is to use TCNs as online learning models[4] for probabilistic forecasting, refining the networks with operational data. For example, a pre-trained TCN could look at past near misses during operation and adjust its weights, improving future predictions.

An advantage related to deep TCN is that external factors can be included in the prediction through the covariates. In the case of DPRA, the events leading up to an end state are only correlated if we consider the context in which they happen. To illustrate, consider a ship navigating in a port, which may collide with a quay at some point in the future. The collision accident event could be preceded by “turn to port” or “increase speed”, but it is hard to see the correlation between the collision and its preceding events if we do not know that these events happened in the context of a ship navigating in a quay. In other words, turning the ship or changing its speed does not necessarily lead to a collision every time, such that the operating context is important. The covariates in deep TCN's probabilistic forecasting could allow for the inclusion of contextual information into the prediction.

3.2. Possible Challenges and Caveats

The fact that TCN and deep TCN are supervised learning methods is simultaneously an advantage, as discussed above, and a disadvantage. One challenge with supervised learning is obtaining a large enough training set to ensure that the network has a good generalization power, and validation and testing sets must also be obtained to verify that the model is correct and generic. To train TCNs for assisting DPRA, these datasets must also be related to accident scenarios specific to the dynamic system being assessed. Popular repositories for machine learning datasets such as Kaggle[5] and the UCI machine learning repository[6] may contain an abundance of datasets for accidents involving a specific type of dynamic system, i.e., traffic accidents with cars, but a scarcity of accident datasets for another type, i.e., ship collisions. In the latter case, a labeled accident dataset would need to be built, which is also a challenge due to its complicated and time-consuming nature.

While the idea of using TCNs to assist DPRA is promising, transparency and explainability are essential for understanding how autonomous systems make decisions based on risk [26]. The former means that a user must be able to see all the events that lead from an initiating event to a possible consequence. The latter means there should be a clear causal link in the sequence of events, i.e., a person should be able to understand how the accident happened from start to end. In general, "black box" approaches for prediction, e.g., learning models which encode patterns and provide only an output, are ill-suited for DPRA since it is difficult to know how the predictions are reached. Therefore, a caveat to using TCN to assist DPRA is that these two requirements must be kept, meaning TCNs should complement DPRA instead of replacing it.

4. CONCLUSIONS

In this paper, an approach was proposed to use temporal convolutional networks to assist DDET-based DPRA, with the main objective of improving the computational performance of the DPRA process. The approach consists of employing TCNs when generating the DDET, where the events leading up to the present time are used as historical data to predict the probability distribution of future accident events. The approach argues that a DDET can be seen as a set of time series data and can thus be framed in the context of probabilistic forecasting problems, which is solved by TCNs. Deep TCN was discussed as an interesting framework for probabilistic forecasting, as it allows for the inclusion of external factors in the prediction through covariates. In the case of DPRA, the covariates could represent contextual information, e.g., operating area.

The main challenge with the approach is related to the fact that TCNs are supervised learning models, and therefore must be trained, validated, and tested. Datasets for supervised learning usually are massive, depending on the task at hand. For example, datasets for object recognition training typically contain millions of images. The dataset for training deep TCN in this context should have accident scenarios specific

---

[4] Learning models which are continuously trained and improved with new examples and data [19].
[5] https://www.kaggle.com/
[6] https://archive.ics.uci.edu/ml/index.php

to the dynamic system being assessed, labeled with events and timestamps. A challenge is to find such a dataset or build one if it does not exist.

As future work, a search for an accident dataset will be performed. Assuming the dataset is found, the deep TCN model will be trained, tuning its parameters to ensure the best model possible. Finally, the accident predictor TCN will be integrated with a method for DDET-based DPRA such as KPRA, as described in [24].

Acknowledgements

This work is sponsored by the Research Council of Norway (RCN) through project number 342641/L10, the RCN Centre of Excellence funding scheme, NTNU AMOS, project number 223254, and the RCN SFI Autoship project, number 309230.

References

[1] M. Rausand and S. Haugen, *Risk Assessment: Theory, Methods and Applications*, 2nd ed. New Jersey, USA: John Wiley & Sons, Ltd, 2020.
[2] I. B. Utne, A. J. Sørensen, and I. Schjølberg, “Risk Management of Autonomous Marine Systems and Operations,” in *Proceedings of the 36th International Conference on Ocean, Offshore and Arctic Engineering*, 2017, pp. 1–10.
[3] R. G. Maidana, T. Parhizkar, A. Gomola, I. B. Utne, and A. Mosleh, “Supervised dynamic probabilistic risk assessment: Review and comparison of methods,” *Reliab. Eng. Syst. Saf.*, vol. 230, p. 108889, 2023.
[4] A. Hakobyan *et al.*, “Dynamic generation of accident progression event trees,” *Nucl. Eng. Des.*, vol. 238, no. 12, pp. 3457–3467, 2008.
[5] Y. Chen, S. Yang, and W. Men, “Automatic Generation of Failure Mechanism Propagation Scenario via Guided Simulation and Intelligent Algorithm,” *IEEE Access*, vol. 7, pp. 34762–34775, 2019.
[6] Y. Hu, T. Parhizkar, and A. Mosleh, “Guided simulation for dynamic probabilistic risk assessment of complex systems: Concept, method, and application,” *Reliab. Eng. Syst. Saf.*, vol. 217, p. 108047, 2022.
[7] A. R. Patel and P. Liggesmeyer, “Machine Learning based Dynamic Risk Assessment for Autonomous Vehicles,” in *Proceedings of the International Symposium on Connected and Autonomous Vehicles*, 2021, pp. 1–7.
[8] P. Feth, M. N. Akram, R. Schuster, and O. Wasenmüller, “Dynamic Risk Assessment for Vehicles of Higher Automation Levels by Deep Learning,” in *Proceedings of the International Conference on Computer Safety, Reliability, and Security*, 2018, pp. 535–547.
[9] Y. Wang and J. Kato, “Collision Risk Rating of Traffic Scene from Dashboard Cameras,” in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications*, 2017, pp. 1–6.
[10] A. Tritsarolis, B. Murray, N. Pelekis, and Y. Theodoridis, “Collision Risk Assessment and Forecasting on Maritime Data,” in *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, 2023.
[11] J. B. Dugan, S. J. Bavuso, and M. A. Boyd, “Dynamic fault-tree models for fault-tolerant computer systems,” *IEEE Trans. Reliab.*, vol. 41, no. 3, pp. 363–377, 1992.
[12] H. Boudali and J. B. Dugan, “A new Bayesian network approach to solve dynamic fault trees,” in *Proceedings of the Annual Reliability and Maintainability Symposium*, 2005, pp. 451–456.
[13] R. Sterritt, A. H. Marshall, C. M. Shapcott, and S. I. McClean, “Exploring dynamic Bayesian belief networks for intelligent fault management systems,” in *Proceedings of the IEEE international conference on systems, man and cybernetics*, 2000, pp. 3646–3652.
[14] M. Marseguerra and E. Zio, “Nonlinear Monte Carlo reliability analysis with biasing towards top event,” *Reliab. Eng. Syst. Saf.*, vol. 40, no. 1, pp. 31–42, 1993.
[15] M. Marseguerra, E. Zio, and F. Cadini, “Biased Monte Carlo unavailability analysis for systems with time-dependent failure rates,” *Reliab. Eng. Syst. Saf.*, vol. 76, no. 1, pp. 11–17, 2002.
[16] T. Parhizkar, I. B. Utne, J. E. Vinnem, and A. Mosleh, “Dynamic probabilistic risk assessment of decision-making in emergencies for complex systems, case study: Dynamic positioning drilling unit,” *Ocean Eng.*, vol. 237, p. 109653, 2021.
[17] K.-S. Hsueh and A. Mosleh, “The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants,” *Reliab. Eng. Syst. Saf.*, vol. 52, no. 3, pp. 297–314, 1996.

[18] Y. Hu, H. Nejad, D. Zhu, and A. Mosleh, "Solution of Phased-Mission Benchmark Problem Using the SimPRA Dynamic PRA Methdology," in *Proceedings of the 8th International Conference on Probabilistic Safety Assessment & Management (PSAM)*, 2006, pp. 1–9.
[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[20] Y. Jiaxin, W. Fang, and Y. Jieru, "A review of action recognition based on Convolutional Neural Network," *J. Phys. Conf. Ser.*, vol. 1827, no. 1, p. 012138, Mar. 2021.
[21] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks: A Unified Approach to Action Segmentation," in *Proceedings of the ECCV Workshop on Computer Vision*, 2016, pp. 47–54.
[22] R. Dai, S. Xu, Q. Gu, C. Ji, and K. Liu, "Hybrid Spatio-Temporal Graph Convolutional Network: Improving Traffic Prediction with Navigation Data," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3074–3082.
[23] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Probabilistic forecasting with temporal convolutional neural network," *Neurocomputing*, vol. 399, pp. 491–501, 2020.
[24] R. G. Maidana, T. Parhizkar, G. S. Martin, and I. B. Utne, "Dynamic probabilistic risk assessment with K-shortest-paths planning for generating discrete dynamic event trees," *Reliab. Eng. Syst. Saf.*, vol. 242, p. 109725, 2024.
[25] M. Cerliani, "Neural Networks Ensemble." Accessed: May 15, 2024. [Online]. Available: https://towardsdatascience.com/neural-networks-ensemble-33f33bea7df3
[26] C. A. Thieme, A. Morozov, S. Blindheim, and R. Maidana, "Artificial Intelligence and Data Analytics in Resilient Autonomous Systems: Report of the Discussions of Breakout Session," in *Proceedings to the International Workshop on Autonomous Systems Safety*, 2021, pp. 36–41.