Simulation Based Machine Learning For Fault Detection In Complex Systems Using The Functional Failure Identification And Propagation Framework

Conference Paper · August 2014

DOI: 10.1115/DETC2014-34628

4 authors:

Nikolaos Papakonstantinou
VTT Technical Research Centre of Finland

Scott Proper
Oregon State University

Bryan O'Halloran
Naval Postgraduate School

Irem Y. Tumer
Oregon State University

Proceedings of the ASME 2014 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2014
August 17-20, 2014, Buffalo, New York, USA

DETC2014-34628

SIMULATION BASED MACHINE LEARNING FOR FAULT DETECTION IN COMPLEX SYSTEMS USING THE FUNCTIONAL FAILURE IDENTIFICATION AND PROPAGATION FRAMEWORK

Nikolaos Papakonstantinou
Department of Electrical Engineering and Automation
Aalto University
FI-00076 AALTO, Espoo, Finland
nikolaos.papakonstantinou@aalto.fi

Scott Proper
Oregon State University
Corvallis, Oregon, 97331 USA
proper@eecs.oregonstate.edu

Bryan O'Halloran
Raytheon Missile Systems
1151 E Hermans Rd, Tucson, AZ 85756
Bryan.M.OHalloran@raython.com

Irem Y. Tumer
School of Mechanical, Industrial, and Manufacturing Engineering
204 Rogers Hall
Oregon State University
Corvallis, Oregon, 97331 USA
irem.tumer@oregonstate.edu

ABSTRACT

Fault detection and identification in mechatronic systems with complex interdependencies between subsystems is a very active research area. Various alternative quantitative and qualitative methods have been proposed in the literature for fault identification on industrial processes, making it difficult for researchers and industrial practitioners to choose a method for their application. The Functional Failure Identification and Propagation (FFIP) framework has been proposed in past research for risk assessment of early complex system designs. FFIP is a versatile framework which has been extended in prior work to automatically evaluate sets of alternative system designs, perform sensitivity analysis, and event trees generation from critical event scenario simulation results. This paper's contribution is an FFIP extension, used to generate the training and testing data sets needed to develop fault detection systems based on data driven machine learning methods. The methodology is illustrated with a case study of a generic nuclear power plant where a fault or the location of a fault within the system is identified. Two fault detection methods are compared, based on an artificial neural network and a decision tree. The case study results show that the decision tree was more meaningful as a model and had better detection accuracy (97% success in identification of fault location).

INTRODUCTION

The identification of faults in complex systems is a difficult task for human operators. Systems are becoming increasingly complex with a rising number of components and interdependencies between the mechanical, electrical and software aspects. During fault situations, this complexity can lead to confusion about the initiating event of a critical event scenario and the location of the fault.

A failure at Three Mile Island is an example of how fault detection is challenging. During this event, one pressure release valve failed open during an emergency shutdown and caused a partial meltdown and destruction of a reactor. The damage was extensive because the fault was not identified early enough by the operators due to inadequate control room instrumentation and operator training programs [1, 2].

The importance of fault detection in complex processes has led to the development of several methods which can be used as the basis for fault detection systems. The research of this paper focuses on data-driven quantitative machine learning methods for fault detection, such as artificial neural networks and decision trees [3]. These methods do not use knowledge about the structure or logic of the system, but instead are “trained” to detect faults using data sets with system variables describing the behavior of the system. These data sets originate from real historical process data, or if real data are not available, data generated by a simulated model. When such a method is trained to give good results, then it is evaluated with a “test” data set.

The contribution of this paper is the extension of the Functional Failure Identification and Propagation (FFIP) methodology to generate training and testing data sets using functional failure results. These sets can be used as input for the development and comparison of fault detection systems based on machine learning methods.

The simulation based framework proposed in this paper is targeting the identification of a large number of faults, when there are no adequate historic data for training. The fault identification systems developed and tested with this method are designed to provide decision making assistance to complex system operators.

LITERATURE REVIEW

This section presents a brief overview of the state of the art in fault detection and diagnosis methods. The research efforts on this field are extensive and a wide variety of alternative fault detection methods and hybrids have been proposed. The contribution of this paper is a methodology for compiling training and testing data sets, using the Functional Failure Identification and Propagation (FFIP) framework, to support a simulation based framework for training and testing alternative methods machine learning based methods for fault detection.

Machine learning studies algorithms enabling computers to learn from data [4]. Although machine learning has been a very active research field since the development of the first artificial neural networks [5], it has advanced significantly over the last decade because of the increased availability of computing power and the development of new methods [6] with many engineering applications.

The abundance of literature on the fault detection and diagnosis domain has motivated researchers to review and categorize the methods, and thus facilitate academic researchers and industrial practitioners. A three part comparative study [3, 7, 8] of literature related to fault detection classifies the methods into three categories. The "quantitative model-based" methods use models of the system to analytically identify inconsistencies between the actual and the expected behavior and then decision rules are applied to perform the diagnosis [7]. The "Qualitative models and search strategies" category includes methods based on non-quantitative models of the system, like fault trees and topographic templates created using expert knowledge [8]. The "Process history based methods" do not use any a priori knowledge about the system, but instead rely on large data sets of process data to enable qualitative (e.g. expert systems) or quantitative (e.g. Artificial Neural Networks) methods [3]. The literature review in [9] focuses model based methods, [10] presents a review of artificial intelligence based methods and [11] presents a benchmark of quantitative statistical data-driven methods while [12] reviews methods applied on transient identification in Nuclear Power Plants (NPPs).

Applications of Artificial Neural Networks [13] have been proposed for safety critical systems [14] and for building quantitative fault diagnosis systems [15]. Supporting nuclear power processes, ANNs have been suggested for fault diagnosis [16], transient diagnosis [17], accident identification [18] and condition monitoring [19].

Hybrids of ANNs with other methods have also been proposed for fault detection. In [20] an analytical method and an ANN are combined into a hybrid diagnostic system applied a NPP case study. A fuzzy logic diagnostic system is combined with an ANN in [21] for early transient identification in NPPs. [22] uses a modified dynamic ANN and a dynamic neuro-fuzzy network to create a NPP accident diagnosis advisory system. ANNs trained with data processed with Fast Fourier Transformation and with simple time-series are combined in [23] to improve fault recognition and provide tolerance against drifts in the measurements. Statistical control charts are combined with ANNs in [24] for improved fault detection.

Decision trees are another quantitative method used for fault diagnosis [25]. Fault detection applications based on decision trees have been proposed for photovoltaic arrays [26], AC transmission lines [27], power systems [28] and migration paths from fault trees to decision trees for fault detection has been suggested for the International Space Station [29].

Fault detection methods have been proposed, based on data-driven modeling and residual space analysis [30], Independent Component Analysis [31], hidden Markov models [32], optimized fuzzy clustering [33] and Dynamic Case Based Reasoning [34].

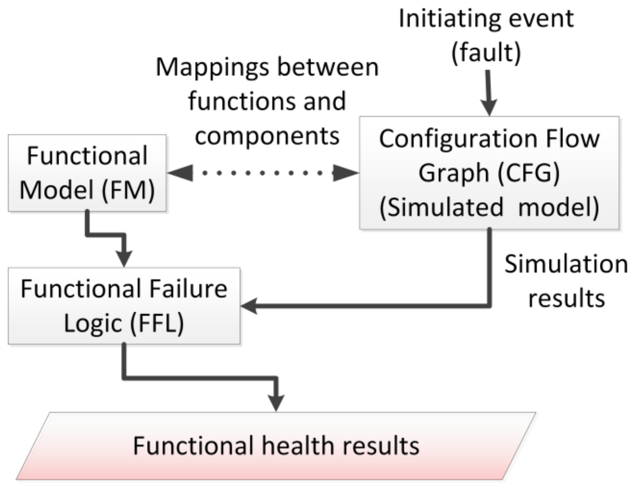

FIGURE 1. Overview of the Functional Failure Identification and Propagation (FFIP) framework.

The Functional Failure Identification and Propagation (FFIP) framework has been proposed in prior work as a risk identification method for early complex system designs. FFIP can identify fault propagation at a functional level across the boundaries of software, electrical and mechanical aspects of the system [35-38]. The method has been extended in past research to support sensitivity analysis [39], alternative system designs [40] and to automatic generate event trees for critical event scenarios [41]. In the methodology presented in this paper FFIP functional failure results are used to generate training and testing input data for process history based quantitative fault detection methods such as ANNs and decision trees.

METHODOLOGY AND TOOLCHAIN

This section describes the methodology to generate sets of data for fault identification. To do this FFIP is used to determine the system's functional health for a set of fault scenarios. To make the method able to identify faults and their locations, we base it on a data driven machine learning algorithm, in our case either an ANN or decision tree. Fault data from the simulations is fed into FFIP and the functional health for that fault is calculated. Over several faults, and several simulations per fault, the method gets trained to predict it.

One aspect of the method is the Functional Failure Identification and Propagation (FFIP) framework [38]. FFIP uses a functional model composed by the system functions. These functions are mapped to process components which contain behavioral logic and are parts of the Configuration Flow Graph. This graph can be simulated and provide results for different initiating events, like component failures. Each function of the functional model contains Functional Failure Logic (FFL) which reasons about the health of the function using simulation results from the Configuration Flow Graph. The functional health results show the impact of the simulated critical event scenario to the functions of the system. An overview of the FFIP framework is presented in Fig. 1.

The second aspect of the method is to use data-driven quantitative fault detection techniques [3], such as artificial neural networks (ANNs) and decision trees. To accurately identify faults, these techniques require being trained and tested with data from the process. The data used for training and testing are generated by process monitoring of key variables (e.g. pressures, temperatures) when faults are present. If historical data from the process are not available, then simulated results of a process model can provide the necessary data. Further, during design activities an adequately built simulation model is often preferred over historical data due to uncertainties in the data. For example, data would be used from similar designs that have different geometry, were built to different specifications, governed by a different budget, etc. A simulation model will more accurately reflect the current design.

Fig. 2 shows the process for training and testing a data-driven quantitative fault detection system. For a single fault, more than one entry in the data set is needed for successful training and testing. This can be accomplished if the fault has happened multiple times during the past, or by performing

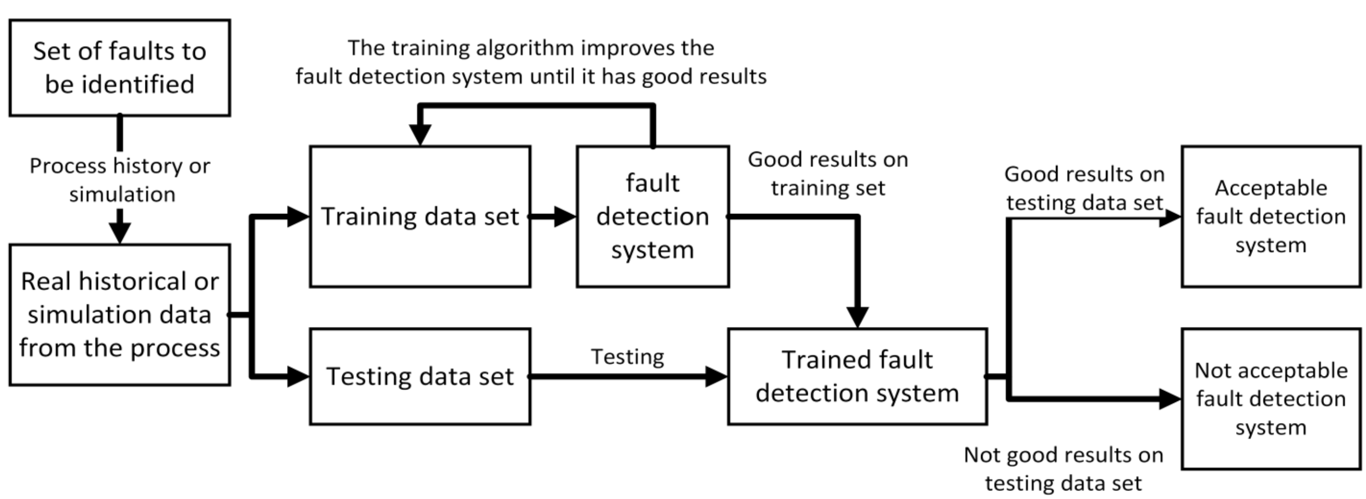

FIGURE 2. Workflow for developing (training and testing) a data-driven fault detection system. The simulation data set is split in order to obtain separate training and testing data sets. This practice ensures that the system is not trained to remember specific data.

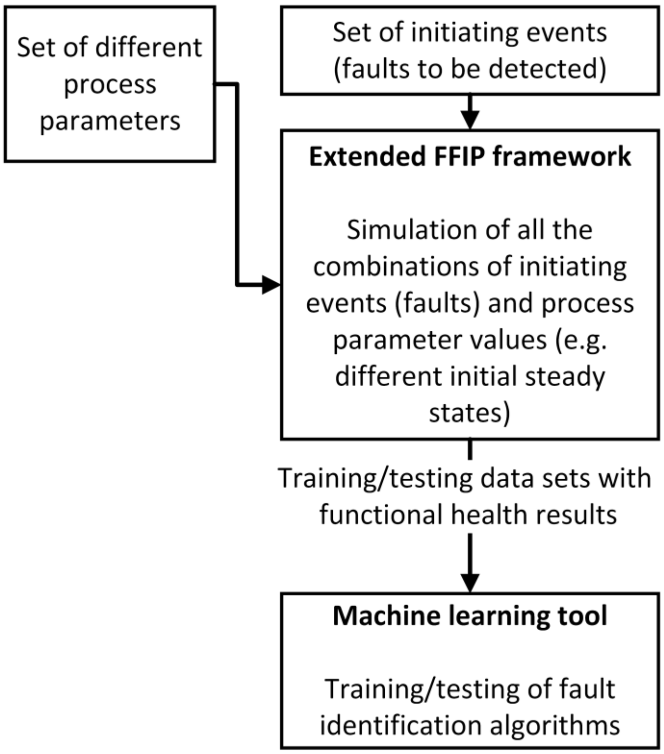

FIGURE 3. A data set generation workflow using the Functional Failure Identification and Propagation framework.

multiple simulations per fault with different simulation parameters. In general, every entry in the data set contains some monitored system variable values and a "classification" attribute. The monitored system variables can be related to the process status (e.g. the temperature of a tank or the mass flow in a pipe). The classification attribute determines what the fault detection system will actually detect. As classification attribute we can use the fault's name, the location of the fault, the fault's type (e.g. electrical, mechanical), etc. The data set is split into two sets, one used for training the fault detection system (training data set) and the other for evaluating its performance (testing data set) [13].

From the pool of all the possible faults, only the faults which affect the system's behavior in the state we are interested in (e.g. in steady state) can be potentially detected by a data driven fault detection system. The reason for the "silent", non-detectable, failures is that some component faults will not affect the process if there is no change in the state of the component because of the fault (e.g. if a valve fails shut when it was shut anyway or if a fault happens in an inactive system). In this case the fault is not detectable by monitoring the process variables.

In this research the FFIP framework is extended to facilitate the development of data-driven quantitative fault detection methods by providing the training and testing data sets. The workflow presented in Fig 3 starts with the preparation of a set with the component faults which should be detected by the fault detection system. Also, to increase the size of the data set, the faults are simulated multiple times, using different process parameter values (e.g. for a power generation process, the output power can be such a parameter). The functional health results for all the simulations are then used to compile the training and testing data sets.

The extension of the FFIP framework is presented in more detail in Fig. 4. The Configuration Flow Graph is simulated for every pair of fault and process parameters. This simulation provides a time series of the monitored process variables (e.g. temperatures, pressures, flows). The simulation results are used by the FFL to generate the functional health results.

The functional health result is composed by three statistical values per monitored signal connected to the FFL. These values are the maximum positive deviation from Steady State (SS) average divided by the SS average, the maximum negative deviation from the SS average divided by the SS average and

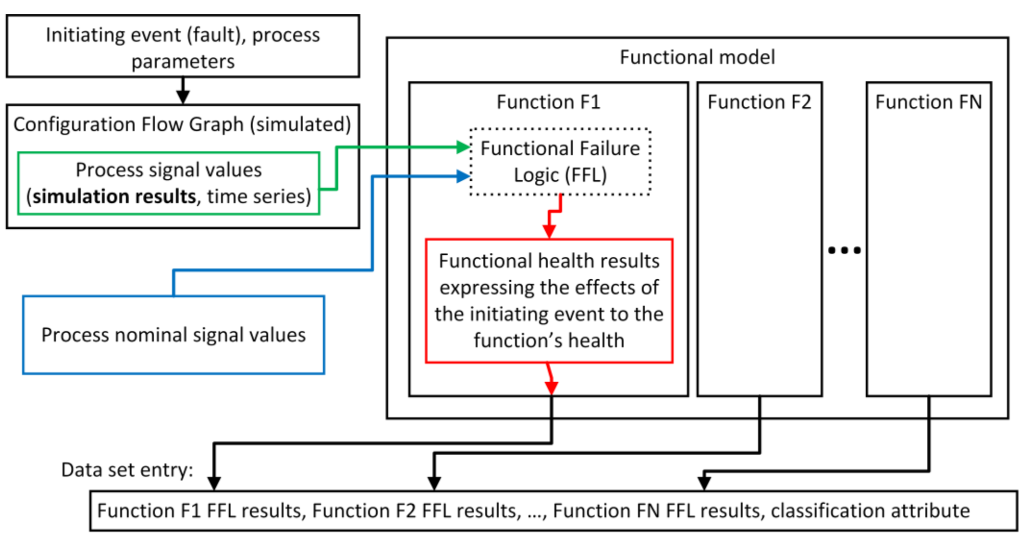

FIGURE 4. Overview of the proposed methodology, the functional failure logic is used to generate data set entries.

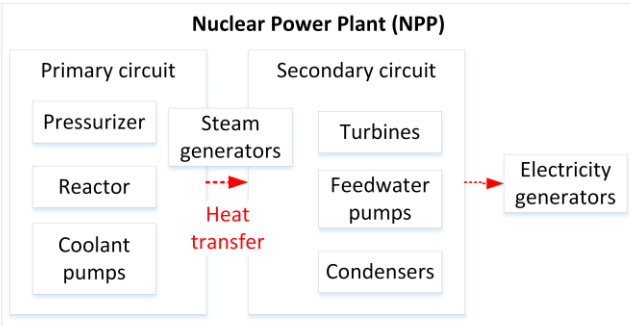

FIGURE 5. A simple model of a typical electricity generation process using nuclear energy.

the maximum deviation of average signal value from the SS average divided by the SS average. All the functional health result values are expressed as percentages. High percentages indicate high impact to the function in that simulation scenario. The functional results for every function of the functional model are serialized and the classification attribute is added to the end.

These results abstract the effects of the critical event scenario to the function over the whole simulation run. The functional health results and the classification attribute are serialized into one data set entry. The complete data set is built when the functional health data set entries for all the pairs of faults and process parameter values are available.

CASE STUDY

The case study for demonstrating the proposed methodology is the development and comparison of two fault detection systems based on data-driven quantitative fault detection methods. The first system uses an ANN and the second is based on a decision tree model. These two systems were developed and tested on a generic nuclear power plant model provided by Fortum Power and Heat [42], a power company operating nuclear power plants in Finland. The model runs on Apros 6, a first principles dynamic process simulator [43] developed by Fortum and VTT Technical Research Centre of Finland [44]. Details of the generic nuclear power plant model, for example component names and component attribute names, will not be published in this article due to proprietary concerns.

The nuclear power plant model is shown in Fig. 5. Two main loops, the primary and secondary circuit are the base for producing electricity using nuclear energy. The fission within the nuclear fuel in the reactor located in the primary circuit generates heat which is transferred to the water in the reactor vessel. The primary coolant pumps circulate water through the reactor and the steam generators. This water flow transfers heat from the reactor vessel to the secondary circuit through the steam generators. The pressurizer, located also in the primary circuit, is a vessel directly connected to the reactor and it is partially filled with water. It is designed to absorb temperature and pressure transients by using heaters and water sprays to

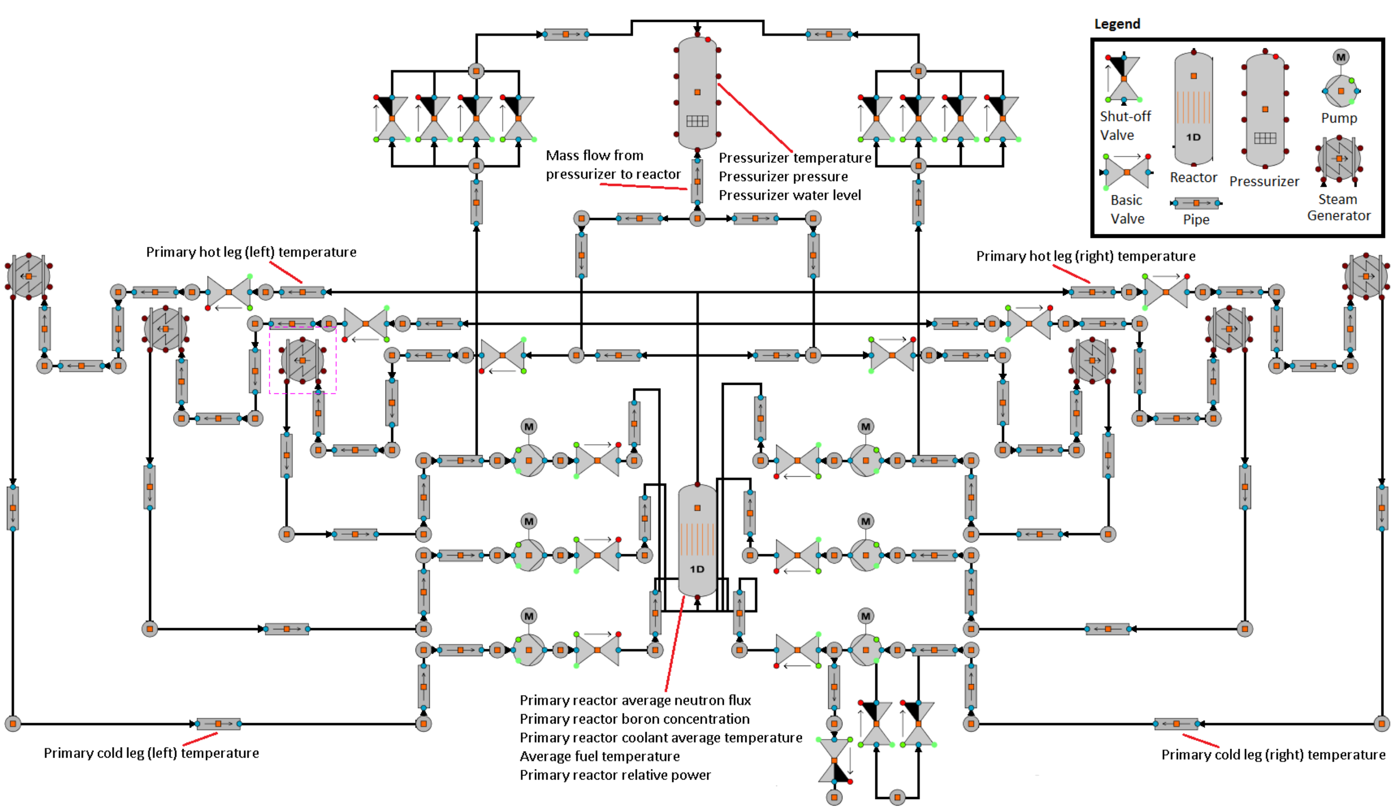

FIGURE 6. The simulation model (configuration flow graph) for the primary circuit of the generic nuclear power plant model in the Apros 6 simulator (modified for readability). Monitored process signals are noted on the diagram.

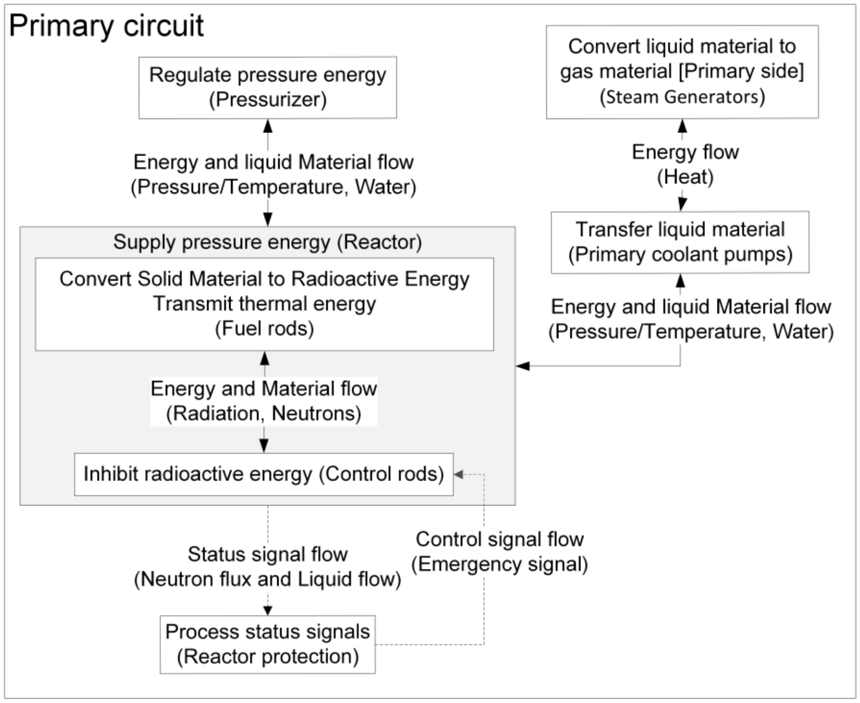

FIGURE 7. Functional model for the primary circuit of the nuclear power plant model. In parenthesis are the components which are mapped to the functions.

keep the water in the reactor vessel under high pressure and prevent boiling. Fig. 6 shows a simplified version of the primary circuit diagram, in the Apros 6 simulator. A high level functional model of the primary circuit with the main functions, mappings to process components, and the energy, material and signal flows between the functions, is shown in Fig. 7.

In the secondary circuit, water flowing through the steam generators is converted to steam. The high pressure steam flows through turbines connected to electric generators. The steam is converted back to water in condensers. There is no direct connection between the primary and secondary circuits apart from the heat transfer in the steam generators.

From the nuclear power plant model, a set of 116 automation components were selected to be the pool of potential faults. These components were primarily valve and pump actuator controllers. Three failure modes were chosen for each automation component type (e.g. a pump actuator controller can be triggered to the “failed stop”, “failed start” or “no electric supply” failure modes which results in stopping, starting or stop controlling the pump). Faults are presented as component – failure mode pairs (e.g., “Pump pumpID” – “Fails Stopped”). From the pool of 348 total faults, only 92 faults affect the power plant model when it runs in steady state and nominal mode and can be potentially detected by a data driven fault detection system. The 256 other faults are “silent” and have no effect to the process during steady state operation.

Multiple simulations per fault are necessary to generate a useful data set. In our case study 11 different steady state power production levels were used for the simulations (from 90.9% plant power output to 100%), see Fig. 8. The combinations of

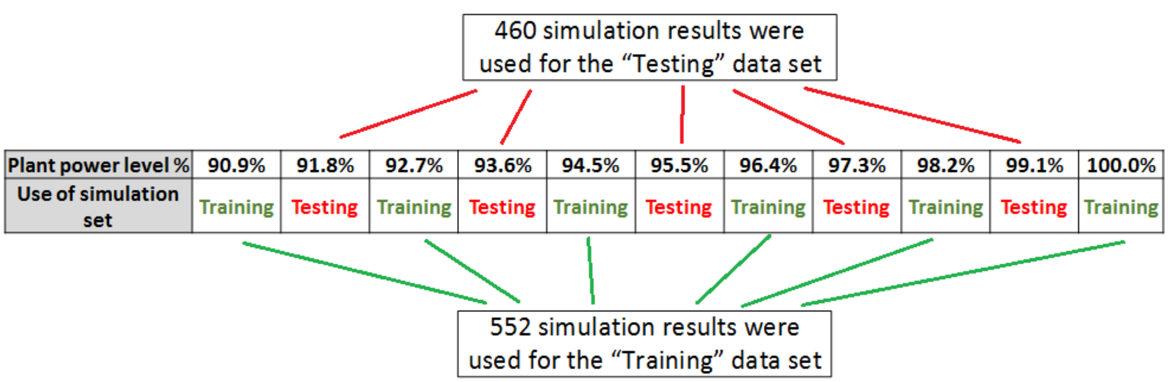

FIGURE 8. The generated data sets. All faults were simulated 11 times, for different steady state power generation levels. The data set entries generated by the functional failure logic where split into a training and a testing data set.

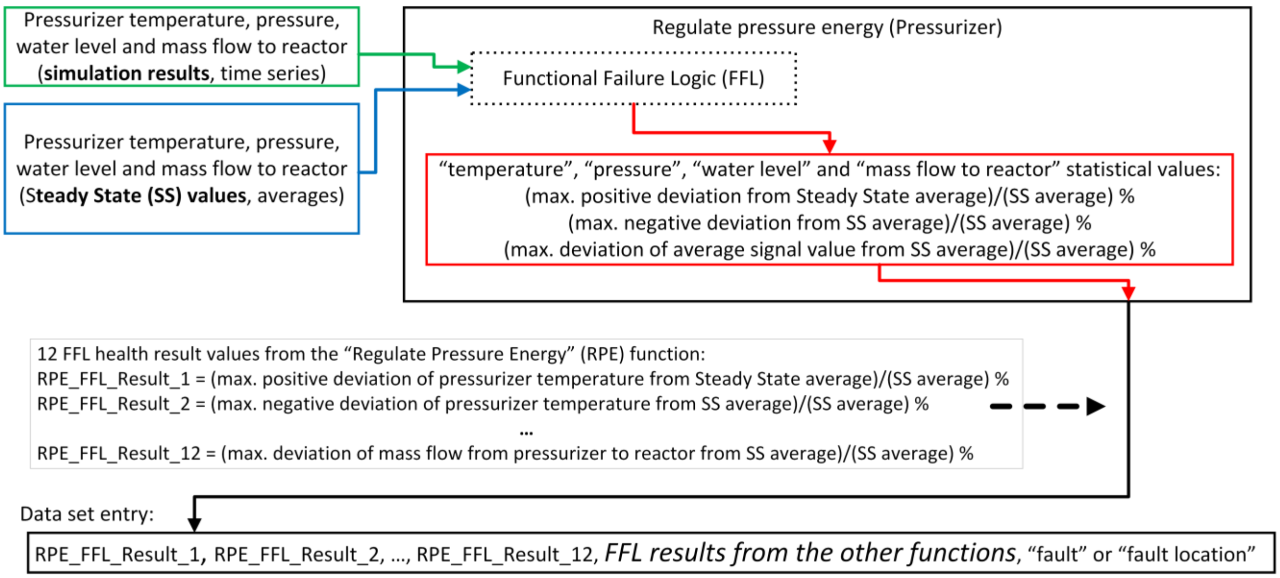

FIGURE 9. An application of the proposed methodology (see Fig. 4). The functional failure logic of the "Regulate pressure energy" functions accepts three signals as inputs (180s time series) and generates three statistical values per signal, to be used for the data set entry.

the 92 detectable faults and the 11 power levels (a total of 1012 simulation scenarios) were used to create the data sets for developing the fault detection systems. The simulations were performed using the Simulation Server component of Apros 6, developed by VTT [44]. Each simulation runs for a simulation time of 180 seconds.

The process for generating entries to the training and testing data sets using the simulation data is shown in Fig. 9.

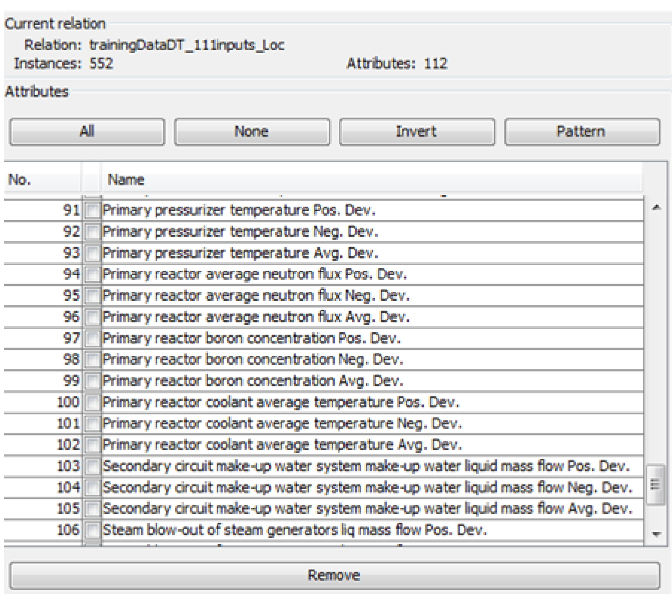

FIGURE 10. List of data set attributes (functional failure results) after importing the training data set for failure location identification to Weka.

The Functional Failure Logic (FFL) for every function of the process uses one or more signals from the simulation results, as well as steady state reference values for these signals, to calculate the functional health result.

As an example, the FFL for the function "Regulate Pressure Energy" in the primary circuit (see Fig. 7), mapped to the pressurizer component in the simulation model (see Fig. 6), uses the "pressurizer temperature", "pressurizer pressure", "pressurizer water level" and "mass flow from pressurizer to reactor vessel" signals as inputs. The functional health result of this function is composed by three statistical values per monitored signal connected to its FFL (9 total values). For the "pressurizer temperature" signal these values are the "pressurizer temperature" maximum positive and negative deviation from Steady State (SS) average divided by the SS average and the maximum deviation of the average "pressurizer temperature" value from the SS average divided by the SS average. The functional results for every function of the functional model (e.g. 12 values for the "Regulate Pressure Energy" function) are serialized and the classification attribute is added to the end. In this case study there were two versions of the data set, one which used the fault (component – failure mode pair) as classification attribute and one which used the fault's location (from a total of 9 possible locations, the main plant systems), see Fig. 9.

A practical issue was the parsing of the simulation results for all the faults, the calculation of the FFL results and finally the generation of the data files to be used for developing the fault detection systems. A software tool was developed to parse all the simulation result files (a total of 1012 files) generated by the simulation server. The tool reads each one of the simulation

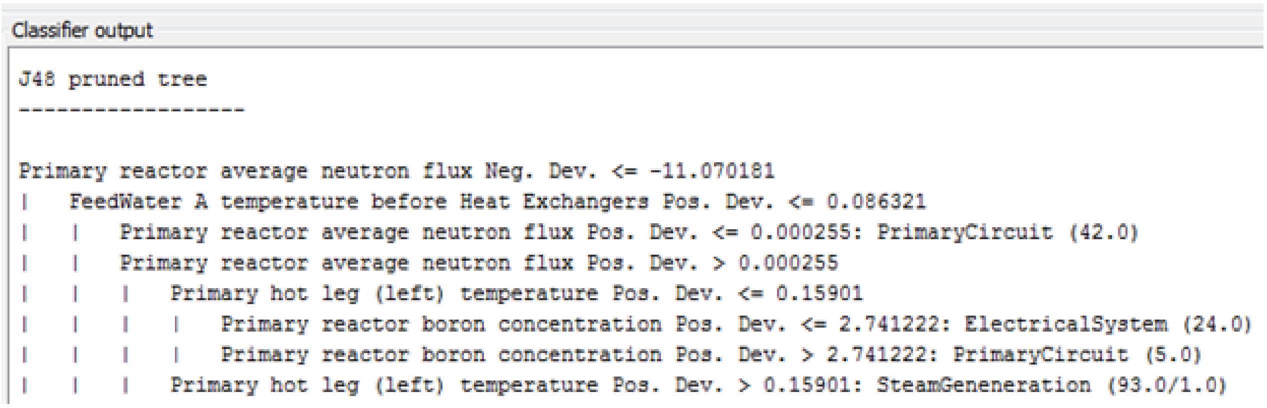

FIGURE 11. Part of the decision tree developed in Weka for fault location identification. Threshold values of the functional failure results are used to determine the fault location.

results files, calculates the functional health results and populates the resulting data set.

The complete resulting data set had 1012 entries, each containing 111 functional health results generated by the FFL (related to 37 monitored signals) and the classification attribute. This data set was split as shown in Fig. 8 to create the training and testing data sets. The FLL results for six plant power output levels were used to build the training data set and the results for five power levels were used for the testing data set.

The WEKA tool [45] was used to train and test the fault detection systems. WEKA is developed by the Machine Learning Group at the University of Waikato and it contains a set of machine learning algorithms for data mining applications. The training and testing data sets with the FFL results were imported to the WEKA tool (see Fig.. 10) and two fault detection methods where tested. Using WEKA's defaults settings a multi layer perceptron ANN [13] and a decision tree were trained and tested. The feedforward ANN had only two layers (input and output) and was trained using back propagation with momentum [13]. The decision tree was based on the J48 algorithm, an open source implementation of the C4.5 algorithm [46], see Fig. 11. The training was for fault identification (92 possible faults) and for fault location identification (9 possible locations - systems).

The comparison of the two methods can be based on the results in fault detection and on other aspects of the methods, such as readability of the generated models and performance during training and testing.

Using the testing data set, the accuracy in fault detection was 64% for the ANN and 82% for the decision tree. For detecting the fault's location (a much easier task since there are only 9 possible fault locations), the artificial neural network had a 93% success rate while the decision tree had 97% success rate (see Fig. 12). The ANN took longer to train than the decision tree, while testing was fast for both.

Apart from accuracy performance and training speed, the decision tree has an advantage in terms of readability. As an example, in Fig. 11 we can understand by reading the first branch of the tree that if the "Primary reactor average neutron flux Negative Deviation from average" is less or equal than -11.070181% and the "FeedWater A temperature before Heat Exchangers Positive Deviation" is less or equal than 0.086321% and the "Primary reactor average neutron flux Positive Deviation" is less or equal than 0.000255% then the fault is located in the "Primary Circuit". This "reasoning" can help the developer of the fault detection system to better understand the results of system and identify functional failure results in training data set which are important for the classification.

These accuracy results are provided as an example application of the framework. Further experimentation with the parameters of the methods can lead to improved results.

CONCLUSIONS

The objective of this research is to provide a simulation based methodology for generating the necessary training and

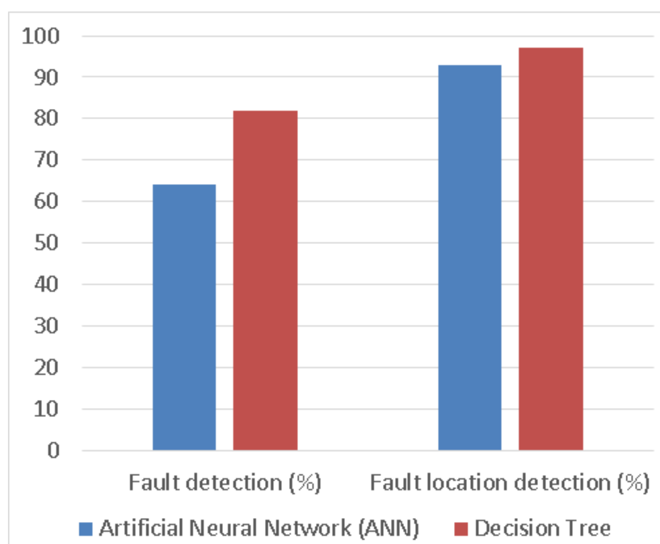

FIGURE 12. Accuracy results for the two fault detection methods. For our case, the decision tree has better accuracy.

testing data sets, using the functional failure results of the FFIP framework. These data sets can be used for the development of machine learning based fault detection systems, targeting complex processes. FFIP's functional failure logic was able to be extended and generate enough functional health results for each function to build adequate data sets.

Although in this research the target was not the optimization of the machine learning algorithms, the two fault detection systems developed in the case study were trained successfully. A comparison of the early results of these two methods showed that the decision tree based fault detection system was trained faster and gave better detection results. Additionally, the decision tree gave a meaningful model which can provide clear reasoning for the classification results of the fault detection system.

Further research can focus on comparing different Functional Failure Logic implementations for the generation of the functional health results. Studying different ways of processing simulation variables time series in order to determine the overall effect to the functional health of a system is an open research question.

The data driven methods used in this paper can benefit by having large data sets, but increasing the data sets can cost in computational time. Further research could study the fault detection accuracy performance in relation to the training and testing data set sizes for different machine learning methods.

This case study focused on the identification of single faults of the automation. Motivated by the increase of computational resources over time, it would be interesting to expand the method towards the identification of combinations of faults and extend the set of faults to the mechanical aspect of the system.

Another research direction is the study of hybrid fault detection systems. A hybrid system which will use one fault detection method to identify the location of a fault and then another method for the identification of the fault itself. Another alternative would be that two or more systems would "vote" and a third system would be trained to use these votes as inputs for fault detection.

The framework presented in this paper has the potential to provide the data sets needed to explore different research paths related to machine learning applications for complex systems. The data set entries are more meaningful when they contain functional failure results. Having a clear methodology for the creation of the necessary data sets can help the development of better fault detection systems for complex processes.

REFERENCES

[1] United States Nuclear Regulatory Commission (NRC). (2013). Three Mile Island Accident. Available: http://www.nrc.gov/reading-rm/doc-collections/fact-sheets/3mile-isle.pdf

[2] American Nuclear Society. (2012). The TMI-2 Accident - In Brief. Available: http://www.new.ans.org/pi/resources/sptopics/tmi/brief.php

[3] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part III: Process history based methods," Computers & Chemical Engineering, vol. 27, pp. 327-346, 2003.

[4] P. Simon, Too big to ignore : the business case for big data.

[5] S. S. Haykin, Neural networks and learning machines, 3rd ed. New York: Prentice Hall, 2009.

[6] F. J. Alexander, "Machine Learning [Guest editor's introduction]," Computing in Science & Engineering, vol. 15, pp. 9-11, 2013.

[7] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," Computers & Chemical Engineering, vol. 27, pp. 293-311, 2003.

[8] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies," Computers & Chemical Engineering, vol. 27, pp. 313-326, 2003.

[9] R. Isermann, "Model-based fault-detection and diagnosis – status and applications," Annual Reviews in Control, vol. 29, pp. 71-85, 2005.

[10] V. Uraikul, C. W. Chan, and P. Tontiwachwuthikul, "Artificial intelligence for monitoring and supervisory control of process systems," Engineering Applications of Artificial Intelligence, vol. 20, pp. 115-131, 2007.

[11] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," Journal of Process Control, vol. 22, pp. 1567-1581, 2012.

[12] K. Moshkbar-Bakhshayesh and M. B. Ghofrani, "Transient identification in nuclear power plants: A review," Progress in Nuclear Energy, vol. 67, pp. 23-32, 2013.

[13] A. K. Jain, M. Jianchang, and K. M. Mohiuddin, "Artificial neural networks: a tutorial," Computer, vol. 29, pp. 31-44, 1996.

[14] Z. Kurd, T. Kelly, and J. Austin, "Developing artificial neural networks for safety critical systems," Neural Computing & Applications, vol. 16, pp. 11-19, Jan 2007.

[15] T. Sorsa and H. N. Koivo, "Application of artificial neural networks in process fault diagnosis," Automatica, vol. 29, pp. 843-849, 1993.

[16] S. G. Vinod, A. K. Babar, H. S. Kushwaha, and V. Venkat Raj, "Symptom based diagnostic system for nuclear power plant operations using artificial neural networks," Reliability Engineering & System Safety, vol. 82, pp. 33-40, 2003.

[17] T. V. Santosh, G. Vinod, R. K. Saraf, A. K. Ghosh, and H. S. Kushwaha, "Application of artificial neural networks to nuclear power plant transient diagnosis," Reliability Engineering & System Safety, vol. 92, pp. 1468-1472, 2007.

[18] T. V. Santosh, A. Srivastava, V. V. S. Sanyasi Rao, A. K. Ghosh, and H. S. Kushwaha, "Diagnostic system for identification of accident scenarios in nuclear power plants using artificial neural networks," Reliability Engineering & System Safety, vol. 94, pp. 759-762, 2009.

[19] E. Ayaz, "Component-wide and plant-wide monitoring by neural networks for Borssele nuclear power plant," Energy Conversion and Management, vol. 49, pp. 3721-3728, 2008.

[20] J. W. Hines, D. W. Miller, and B. K. Hajek, "A hybrid approach for detecting and isolating faults in nuclear power plant interacting systems," Nuclear Technology, vol. 115, pp. 342-358, Sep 1996.

[21] A. Ikonomopoulos, L. H. Tsoukalas, and R. E. Uhrig, "A Hybrid Neural Network - Fuzzy Logic Approach to Nuclear Power Plant Transient Identification," presented at the International conference on frontiers in innovative computing for the nuclear industry, Jackson, Wyoming, USA, 1991.

[22] S. J. Lee and P. H. Seong, "A dynamic neural network based accident diagnosis advisory system for nuclear power plants," Progress in Nuclear Energy, vol. 46, pp. 268-281, 2005.

[23] M. J. Embrechts and S. Benedek, "Hybrid identification of nuclear power plant transients with artificial neural networks," Industrial Electronics, IEEE Transactions on, vol. 51, pp. 686-693, 2004.

[24] R. P. Leger, W. J. Garland, and W. F. S. Poehlman, "Fault detection and diagnosis using statistical control charts and artificial neural networks," Artificial Intelligence in Engineering, vol. 12, pp. 35-47, 1998.

[25] M. Chen, A. X. Zheng, J. Lloyd, M. I. Jordan, and E. Brewer, "Failure diagnosis using decision trees," in Autonomic Computing, 2004. Proceedings. International Conference on, 2004, pp. 36-43.

[26] Y. Zhao, L. Yang, B. Lehman, J. F. de Palma, J. Mosesian, and R. Lyons, "Decision Tree-Based Fault Detection and Classification in Solar Photovoltaic Arrays," 2012 Twenty-Seventh Annual Ieee Applied Power Electronics Conference and Exposition (Apec), pp. 93-99, 2012.

[27] S. R. Samantaray, "Decision tree-based fault zone identification and fault classification in flexible AC transmissions-based transmission line," Generation, Transmission & Distribution, IET, vol. 3, pp. 425-436, 2009.

[28] T. Amraee and S. Ranjbar, "Transient Instability Prediction Using Decision Tree Technique," Power Systems, IEEE Transactions on, vol. 28, pp. 3028-3037, 2013.

[29] C. Lee, R. L. Alena, and P. Robinson, "Migrating Fault Trees To Decision Trees For Real Time Fault Detection On International Space Station," in Aerospace Conference, 2005 IEEE, 2005, pp. 1-6.

[30] B. Lu and B. R. Upadhyaya, "Monitoring and fault diagnosis of the steam generator system of a nuclear power plant using data-driven modeling and residual space analysis," Annals of Nuclear Energy, vol. 32, pp. 897-912, 2005.

[31] A. Ajami and M. Daneshvar, "Data driven approach for fault detection and diagnosis of turbine in thermal power plant using Independent Component Analysis (ICA)," International Journal of Electrical Power & Energy Systems, vol. 43, pp. 728-735, 2012.

[32] K.-C. Kwon and J.-H. Kim, "Accident identification in nuclear power plants using hidden Markov models," Engineering Applications of Artificial Intelligence, vol. 12, pp. 491-501, 1999.

[33] E. Zio and P. Baraldi, "Identification of nuclear transients via optimized fuzzy clustering," Annals of Nuclear Energy, vol. 32, pp. 1068-1080, 2005.

[34] H. R. Berenji and W. Yan, "Case-Based Reasoning for Fault Diagnosis and Prognosis," in Fuzzy Systems, 2006 IEEE International Conference on, 2006, pp. 1316-1321.

[35] S. Sierla, I. Tumer, N. Papakonstantinou, K. Koskinen, and D. Jensen, "Early integration of safety to the mechatronic system design process by the functional failure identification and propagation framework," Mechatronics, vol. 22, pp. 137-151, 2012.

[36] I. Y. Tumer and C. S. Smidts, "Integrated Design and Analysis of Software-Driven Hardware Systems," IEEE Transactions on Computers, vol. 60, pp. 1072-1084, 2011.

[37] T. Kurtoglu, I. Y. Tumer, and D. Jensen, "A functional failure reasoning methodology for evaluation of conceptual system architectures," Research in Engineering Design, vol. 21, pp. 209-234, 2010.

[38] T. Kurtoglu and I. Y. Tumer, "A Graph-Based Fault Identification and Propagation Framework for Functional Design of Complex Systems," Journal of Mechanical Design, vol. 130, 2008.

[39] N. Papakonstantinou, D. Jensen, S. Sierla, and I. Tumer, "Capturing interactions and emergent failure behavior in complex engineered systems and multiple scales," presented at the ASME IDETC/CIE, Washington, DC, USA, 2011.

[40] N. Papakonstantinou, S. Sierla, I. Y. Tumer, and D. Jensen, "Using Fault Propagation Analyses for Early Elimination of Unreliable Design Alternatives of Complex Cyber-Physical Systems," presented at the ASME IDETC/CIE 2012, Chicago, Illinois, USA, 2012.

[41] N. Papakonstantinou, S. Sierla, B. O'Halloran, and I. Y. Tumer, "A Simulation Based Approach to Automate Event Tree Generation for Early Complex System Designs," presented at the ASME IDETC/CIE 2013, Portland, Oregon, USA, 2013.

[42] Fortum. (2014). Fortum Corporation homepage. Available: http://www.fortum.com/en/pages/default.aspx

[43] K. Juslin, "A Companion Model Approach to Modelling and Simulation of Industrial Processes," Doctoral dissertation, Department of Automation and Systems Technology, Helsinki University of Technology, Espoo, 2005.

[44] Technical Research Centre of Finland (VTT). VTT Technical Research Centre of Finland homepage. Available: http://www.vtt.fi/?lang=en

[45] WEKA. (2014). Machine Learning Project at the University of Waikato in New Zealand. Available: http://www.cs.waikato.ac.nz/ml/index.html

[46] J. R. Quinlan, C4.5 : programs for machine learning. San Mateo, Calif.: Morgan Kaufmann Publishers, 1993.

