Smart and Sustainable Manufacturing Systems

doi:10.1520/SSMS20190029 / Vol. 3 / No. 2 / 2019 / available online at www.astm.org

Dy D. Le,[1] Vung Pham,[2] Huyen N. Nguyen,[2] and Tommy Dang[3]

Visualization and Explainable Machine Learning for Efficient Manufacturing and System Operations

Reference

D. D. Le, V. Pham, H. N. Nguyen, and T. Dang, "Visualization and Explainable Machine Learning for Efficient Manufacturing and System Operations," *Smart and Sustainable Manufacturing Systems* 3, no. 2 (2019): 127–147. https://doi.org/10.1520/SSMS20190029

ABSTRACT

To enable Industry 4.0 successfully, there is a need to build a resilient automation system that can quickly recover after having been attacked or robustly sustain continued operations while being threatened, enable an automated monitoring evolution via various sensor channels in real time, and use advanced machine learning and data analytics to formulate strategies to mitigate and eliminate faults, threats, and malicious attacks. It is envisioned that if we can develop an intelligent model that (a) represents a meaningful, realistic environment and complex entity containing manufacturing Internet of Things interdependent and independent properties that are stepping-stones of the cyber kill chain or precursors of the onset of cyberattacks; (b) can learn and predict potential errors and formulate offense/defense strategies and healing solutions; (c) can enable cognitive ability and human-in-the-loop analytics in real time; and (d) can facilitate system behavior changes to disrupt the attack cascade, then the hosting system can learn how to neutralize threats and attacks and self-repair infected or damaged links autonomously. In this article, our preliminary work presents a visual analytics framework and technique for situational awareness, including autonomously monitoring, diagnosing, and prognosticating the state of cyber-physical systems. Our approach, presented in this article, relies on visual characterizations of multivariate time series and real-time predictive analytics to highlight potential faults, threats, and malicious attacks. To validate the usefulness of our approach, we demonstrate the developed technique using various aviation datasets obtained from the Prognostics Center of Excellence at the National Aeronautics and Space Administration Ames.

Keywords

situational awareness visual analytics, human-in-the-loop analytics, visual characterizations, resilient automation system, efficient manufacturing automation, explainable machine learning, cyber-physical systems

Manuscript received August 26, 2019; accepted for publication October 16, 2019; published online November 14, 2019.

[1] Institute for Materials, Manufacturing, and Sustainment, Texas Tech University, 2500 Broadway, Lubbock, TX 79409, USA, https://orcid.org/0000-0003-0409-8511

[2] Department of Computer Science, Texas Tech University, 2500 Broadway, Lubbock, TX 79409, USA, https://orcid.org/0000-0001-9702-8904 (V.P.), https://orcid.org/0000-0001-6554-2327 (H.N.)

[3] Department of Computer Science, Texas Tech University, 2500 Broadway, Lubbock, TX 79409, USA (Corresponding author), e-mail: Tommy.Dang@ttu.edu, https://orcid.org/0000-0001-8322-0014

Introduction

Connected manufacturing devices and systems and digitalization (Industry 4.0) have become a reality and promise transformative results to increase energy efficiency and speed. Industry 4.0 enables effective automation and an efficient supply chain process. In Industrial Control Systems automation, it connects several layers of critical links, including cyber-physical components, systems, networks, and controls. As a result, it presents a higher level of complexity and vulnerability to cybersecurity threats.

In the case of manufacturing automation, the system is treated as a whole with hierarchical components, in which almost any stages and components can be the targets of security threats. Cybercriminals may take advantage of these weaknesses to steal intellectual properties, gain access to propriety information, and cause system disruptions. In the United States (U.S.), the manufacturing sector is the potential target for 35 % of all cyber-espionage attacks as recorded by the Verizon data breach investigation in 2017.[1] Furthermore, when cybercriminals try to break into the systems, the weakest link inside a manufacturing system normally comes from human factors. Malicious attacks can also result in the loss of continued operations of the whole manufacturing network, which can severely compromise the effectiveness of the Industry 4.0 revolution and the U.S. economy's competitiveness in the world.

To enable Industry 4.0 successfully, there is a need to build a resilient automation system that can quickly recover after having been attacked or robustly sustain continued operations while being threatened, enable an automated monitoring capability to capture information from manufacturing Internet of Things (IoT) and automation devices in real time, and use advanced artificial intelligence and data analytics to formulate strategies to mitigate and eliminate malicious threats and attacks. Discovery of threats and detection of attacks require advanced tools and methods to fuse a vast amount of diverse and disparate datasets from complex Industry 4.0 infrastructures and components and intelligently dissect them in such a way to provide insights into the threats. The task can be human assisted. However, as digitalization is becoming fully operational, advanced technologies for automated threat and attack monitoring, discovery, and detection are critically needed for effectively securing Industry 4.0 infrastructures and operations.

CURRENT CHALLENGES FOR SECURING EFFICIENT MANUFACTURING AUTOMATION

Securing efficient manufacturing automation is critical for Industry 4.0 success. On March 26, 2019, the Office of Energy Efficiency and Renewable Energy issued a funding opportunity announcement on Clean Energy Manufacturing Innovation Institute: Cybersecurity in Energy Efficient Manufacturing.[2] The manufacturing community has also opinionated the following challenges:

- They are concerned with the supply chain vulnerability, particularly with small companies. This weak link can easily compromise and disrupt the whole manufacturing automation ecosystem and substantially reduce energy efficiency. Adversaries will try to penetrate those small companies, which, in some cases, may be acquired by larger Original Equipment Manufacturers (OEMs), presenting a bigger cybersecurity problem for the parent company and supply chain.[3]
- Small companies often ignore cybersecurity issues because of cost, time, or both.[4] It is not feasible, economically and timely, for them to enable the cybersecurity capability at an adequate level to effectively defend themselves.[5] The manufacturing community must help small companies by developing cybersecurity tools and methods, which must not be only affordable but also scalable and easy for them to implement.
- While the supply chain cybersecurity problem is enormous, we should not forget manufacturing automation vulnerability.[6]
- Currently, we do not have adequate solutions for real-time monitoring and faster response or mitigation to cybersecurity threats and attacks and how to minimize the entire automation downtime due to attacks.[7]

POTENTIAL SOLUTIONS FOR SECURING EFFICIENT MANUFACTURING AUTOMATION

Subject matter experts on manufacturing cybersecurity have also expressed the following thoughts for securing efficient manufacturing automation:

- Currently, there may not be any workable universal solutions to solve the manufacturing cybersecurity problem even within large manufacturing companies and OEMs.[8] One of the perspectives is to force adversaries to spend a vast amount of time and money when trying to attack, hence discouraging or eliminating their malicious intent.
- Best practice[8] to fight back cybersecurity threats/attacks does not exist and is desperately needed. Current approaches of manufacturing systems organizations are unsystematic and not automated.[7] Standardization also needs to be reviewed and updated for cybersecurity issues.[8]
- We need to build a resilient manufacturing system that can quickly recover and resume operations after having been or while being attacked. We should integrate cybersecurity measures into the system during the design phase.[8,9]
- We need to build a real-time automated monitoring capability to capture information from the manufacturing IoT and automation devices and use data analytics to formulate strategies to mitigate and eliminate cybersecurity threats/attacks.[10,11]
- We need to have a comprehensive test to validate, certify, and patch developed cybersecurity technologies and tools for the manufacturing community.[8,9]
- We need to have strategies for technology transition, commercialization, and outreach to bring cybersecurity awareness to the supply chain community.[12]

MAIN CONTRIBUTIONS OF THIS WORK

To address the challenges and concerns described earlier, we develop a visualization technique that aims at enabling real-time monitoring and situation awareness applicable to manufacturing automation networks and other complex system operations. Our contributions in this work include:

- We propose a framework for machine learning–based diagnostics, prognostics, and situation awareness designed to protect manufacturing networks from cyberattacks. The framework combines the strengths of both components (humans and machines) in the decision-making process.
- Since the framework for securing efficient manufacturing automation covers a variety of advanced technologies, we focus on the link between machine learning and visual analytics. Our approach relies on visual characterizations of multivariate time series[13] and real-time predictive analytics[14] to project potential threats and predict the time to detect malicious attacks and determine the time for the system to fail.
- To validate the usefulness of our approach, we demonstrate our techniques on various aviation datasets obtained from the Prognostics Center of Excellence (PCoE) at the National Aeronautics and Space Administration (NASA) Ames.[15] Our proposed system is scalable with larger data and can be adapted for Industry 4.0 to build a resilient and efficient manufacturing automation system.

This article is organized as follows: in the next section, we introduce our conceptualized framework for securing efficient manufacturing automation. We then present the two main components in the framework, as well as the link between them, through an aviation case study. In the "Discussion of Research Potential Benefits" section, we highlight the potential benefits and scalabilities of the proposed system. Finally, we present conclusions and future work for this research.

Research Discussion

DEVELOPMENT OF CAPABILITY FOR SECURING EFFICIENT MANUFACTURING AUTOMATION FRAMEWORK

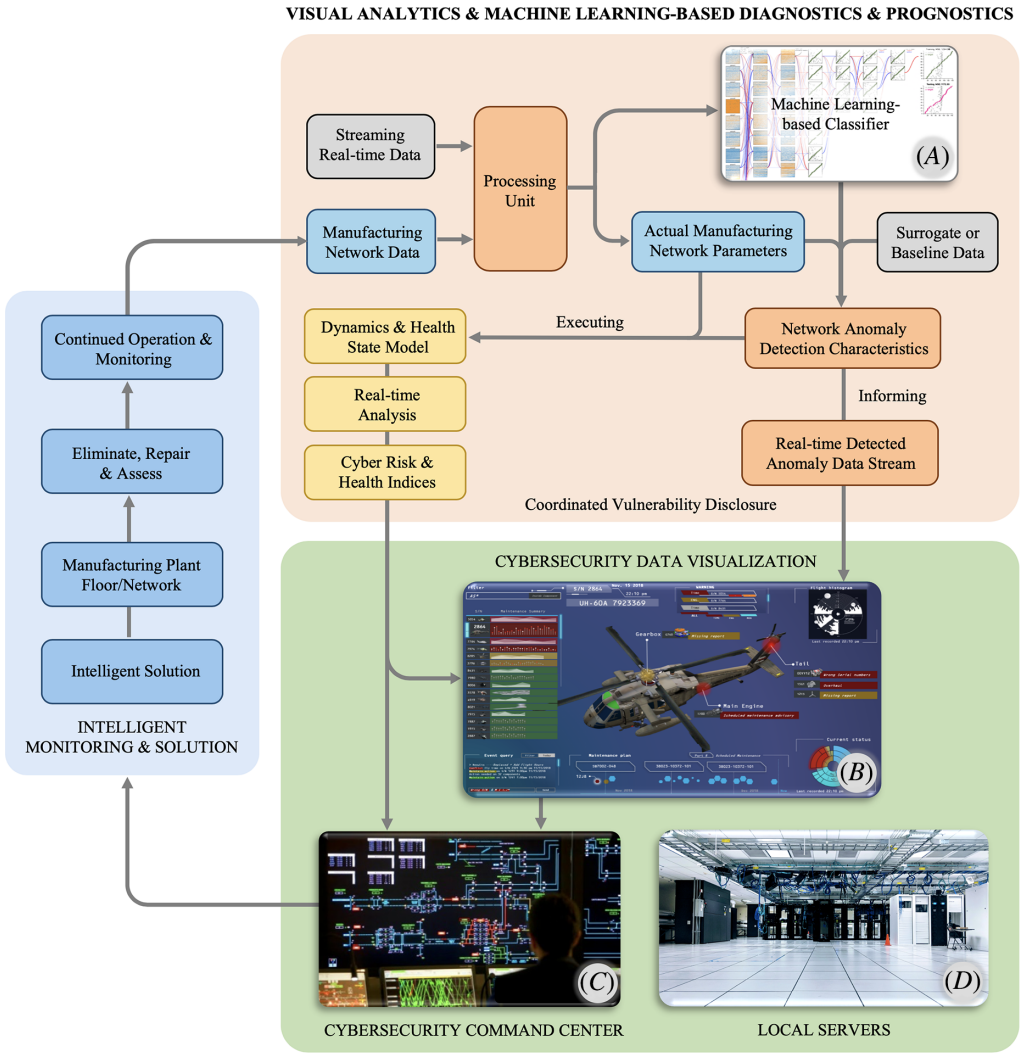
Texas Tech University (TTU) Institute for Materials, Manufacturing, and Sustainment and the Computer Science Department's interactive Data Visualization Laboratory are currently exploring an advanced concept to enable the capability for securing efficient manufacturing automation. The TTU research team has recently established a framework for a proof-of-concept design for a machine learning–based health state awareness system that can potentially be developed for testing and demonstrating its innovation and effectiveness to secure the

manufacturing automation network. The conceptualized framework aims at enabling the real-time monitoring and situation awareness applicable to the manufacturing automation network and other complex system operations.

Figure 1 illustrates a framework for machine learning–based diagnostics, prognostics, and situation awareness designed to protect manufacturing networks from cyberattacks. As raw sensor data and information are being streamed, the embedded machine learning–based system (e.g., hardware and software) is designed to perform the following tasks in real time.

a) Conduct machine learning–based anomaly classifications: Using the streamed and baseline data, the envisioned system conducts the anomaly classification task using machine learning–based training models. If an anomaly were detected, relevant characteristics would immediately be transmitted to operators and stakeholders. At a local manufacturing site, high-performance server systems, as shown in figure 1*D*, are also designed to receive real-time inputs from the network to perform the modeling and analysis using

FIG. 1 Conceptualized framework for securing efficient manufacturing automation. (*A*) Machine learning–based classifier; (*B*) visualization dashboard for health state monitoring; (*C*) cybersecurity command center; (*D*) high-performance local servers.



high-fidelity models and neural network. The results from the machine-learning classification would be automatically used for real-time analysis and visualization to update the existing health model to compute cyber risk and health indexes and, simultaneously, display the network integrity. Analytics visualization techniques and tools are used to intelligently display the health state of the network and relevant information for monitoring and decision-making at the cybersecurity command center.

b) With humans in the loop at the command center, as shown in **figure 1C**, intelligent solutions are autonomously formulated and incorporated with influential parameters (e.g., efficiency, peak load, fuel consumption, setup time, recovery time, minimum energy loss, meantime-to-detect, and other operational parameters, or both), which have a direct relationship between, for example, maximum and minimum efficiency loss.

c) Since the framework for securing efficient manufacturing automation, shown in **figure 1**, may cover a variety of advanced tools and technologies, it is not feasible to address them all. In this article, we will only focus on the analytics capability and functionality using state-of-the-art data mining and visualization techniques, as shown in **figure 1*A*** and **1*B***.

ANALYTICS, VISUALIZATION, AND DATA MINING TECHNIQUES

Since we have not yet obtained appropriate manufacturing automation network data for validation and testing of the developed techniques, we employ a component-wise system modeling approach for a similarly complex system, e.g., aviation maintenance operation, which can later be applied to a typical manufacturing network. Reference performance metrics for our design will include error rates of attack classification, the energy consumption efficiency as a result of the selected security solution, usability of the visualization system, and processing and computation overhead. Performance benchmarking for our techniques is guided by standards specified in the National Institute of Standards and Technology cybersecurity framework.[16]

To develop the visual analytics and validate the whole system, TTU researchers are working in the following specific areas:

(1) Preliminary data gathering and analysis: In this article, we will use aircraft engine datasets, e.g., to represent a complex system operation, obtained from NASA Ames for visualization and validation purposes. We will use these datasets to simulate and categorize threat models of possible vulnerability exploits. Eventually, we will work with our collaborators in the aircraft manufacturing sector to identify and assess manufacturing components and system vulnerabilities as well as collect representative data from their manufacturing systems.

(2) Designing visualization system: This area comprises three major activities[17]: (a) design system software architecture (e.g., formulating requirements, performing use-case analysis, and developing component diagrams); (b) implement a prototype and interactive functionalities of the system; and (c) perform usability tests of visual display and interactive operations at different scales. Our visualization system will be discussed in further detail in the next section.

(3) Building prediction models of threats: For this area, we will develop prediction models (e.g., using deep learning) for the detection of anomalies that can represent a cybersecurity breach. The resulting patterns and models can be used to predict future attacks. On top of the modeling engine, we will finally overlay the visualization framework and support human-in-loop machine learning.[18]

The predictive analytics results of various machine learning techniques (such as Neural Network[19]) and configurations (such as the number of hidden layers and number of neural nodes in each layer) are plotted on the visual interface. Users/operators can visualize and analyze the results and make decisions.[20] Second, intermediate steps of predictive analytics (such as the intermediate results and learned features) are also presented to the users.[21] Users can now inject their cognitive capability, visual reasoning,[22] and domain knowledge into the learning process[23] to further improve the predictive results considering the requirement for energy saving (e.g., trade-off between cost, efficiency, and accuracy). Our research aims to support human-in-loop machine learning and real-time visual analytics.[24] Areas (2) and (3) are an iterative process with humans (e.g., subject matter experts) at the center.

Complex System Candidates for Case Study

CURRENT AND FUTURE AIRCRAFT MANUFACTURING PROCESS

In general, the quality and reliability of aviation platforms while in service rely on many factors, which include having good design, meeting the Federal Aviation Administration regulations and certification guidelines and requirements, and implementing an efficient and streamlined manufacturing process. Advanced technological changes during the past decade, with the use of electronic records and automation, have improved the quality and reliability of aircraft but, at the same time, also introduced serious cybersecurity issues.[25,26]

The aircraft manufacturing process generally covers six significant assemblies: the fuselage, tail assembly, wings, landing gear, engine, and flight controls systems. It is a complex process that not only demands high-quality parts but also smaller dimensional tolerances. Traditionally, aircraft manufacturing adopts similar assembly lines, used by the automotive industry, to create a series of "positions" and "setbacks" to inform the state of the major system and subsystems during the manufacturing process. Many quality control methods are being used, including qualification testing to verify that manufactured components or systems meet the design safety factor.

Because of growing demands in air travel, the traditional assembly lines can no longer keep up with the current demand for higher productivity of aircraft, including parts and accessories. As a result, in order to increase productivity as well as improve energy-consuming efficiency, quality, and smaller dimensional tolerances, aircraft manufacturers have been significantly investing in manufacturing automation (e.g., potentially replace human workers with intelligent robotic systems) embedded at "position" and "setback" locations within the manufacturing process at their assembly plants.

Robotic systems require computerized controls, involving computer-intensive systems, and a substantial amount of knowledge and data to complete a simple task such as riveting. Eventually, aircraft manufacturers will incorporate digital design and electronic record entirely and create integrated assembly line automation. This new and revolutionary manufacturing philosophy will significantly increase productivity, improve aircraft reliability and safety, and reduce manufacturing energy consumption. However, it is also seriously vulnerable to a simple cyberattack if not well protected. For example, the U.S. Army is currently building prototypes for the future vertical lift aircraft for expeditionary military campaigns. As a result, sensitive and highly classified information will be digitally generated during the manufacturing process. Hence, product architecture security and protection must be ensured to eliminate cyber vulnerabilities.

CURRENT AND FUTURE AIRCRAFT MAINTENANCE PROCESSES

The U.S. Army operates the largest fleet of rotorcraft (more than 4,000) in the world. In rotorcraft, there are approximately 200 to 400 flight critical parts whose failure will result in a catastrophic crash. As a result, many critical flight components have been replaced well before having fully consumed the designed life. The U.S. Army imposes a phase time-based maintenance program, which includes preventive maintenance inspections (PMIs). For example, the U.S. Army maintenance team conducts two time periods for PMIs for the Utility Helicopter-60 Black Hawk helicopter. The first PMI is conducted at 360 flight hours and lasts for 7 days. The second PMI is conducted at 720 flight hours and lasts approximately 14 days. These PMIs are required even during platform deployment. When a Black Hawk helicopter returns to its base, it will go through a complete overhaul, causing more downtime and further reducing the aircraft availability. With the introduction of the Health and Usage Monitoring Systems (HUMS), the civil and military rotorcraft communities have begun to move toward the condition-based maintenance in addition to time-based maintenance. HUMS capabilities offer some safety benefits and maintenance credits. However, HUMS technologies may have been saturated and require a substantial improvement in their ability to reliably detect impending mechanical failures or provide credits for reducing scheduled maintenance. From 2009 to 2014, numerous Super Puma helicopters fatally crashed into the North Sea in the United Kingdom because of mechanical problems. Super Puma helicopters, flown in the

North Sea to transport oil workers to and from the oil platforms, were equipped with HUMS. The latest Super Puma helicopter fatal crash occurred on April 1, 2014, killing all 14 passengers and 2 crew on board.[27]

The described challenges and shortfalls, in reference to military operations, result in low optempo (inability to provide critical resources required for the Armed Forces to conduct and support full-spectrum operations training, maintain unit equipment, and sustain routine, day-to-day operations) and the need for large complex maintenance infrastructures and resources. These challenges and shortfalls, if unsolved, will continue to have a significant impact on the U.S. Armed Forces campaigns and expeditionary operations as well as the successful development and execution of the Third Offset Strategy, e.g., series of strategic capabilities that must be developed to give the U.S. forces decisive military technological offsets that generate lasting asymmetrical advantages over any potential adversary for the next 25 to 50 years.

To address those challenges, one of the strategies is to develop component tracking and aircraft health state awareness technologies that facilitate the automation of aircraft records, hence reducing maintenance burden and error. Unfortunately, similar to the case of manufacturing automation, data integrity and security are the major challenges, and they can be detrimentally compromised because of cyber threats, including hacking as well as viruses and malware. The developed framework, as shown in **figure 1**, for securing efficient manufacturing automation can certainly be adopted for securing aviation maintenance automation.

In the aviation case study, we adopt a visual analytics framework, designed for securing efficient manufacturing automation, and use it to demonstrate how data analytics and visualization can help form intelligent solutions for flying the aircraft efficiently and safely under adverse conditions or with detected problems. Specifically, we use engine system datasets obtained from the PCoE at NASA Ames[15] for demonstrating the efficacy of the developed techniques to enable the ability to (a) highlight potential threats (e.g., poor engine performance) and predict the time it takes to have adverse effects (e.g., degraded efficiency); (b) detect malicious attacks (e.g., engine anomalies/irregularities or faults/degradations) and determine the time it takes for the system to fail (e.g., engine failure); and (c) make intelligent recommendations including scheduled maintenance actions to mitigate the detected problems.

VISUALIZATION SYSTEM FOR AIRCRAFT THREAT MONITORING AND MITIGATION ACTIONS

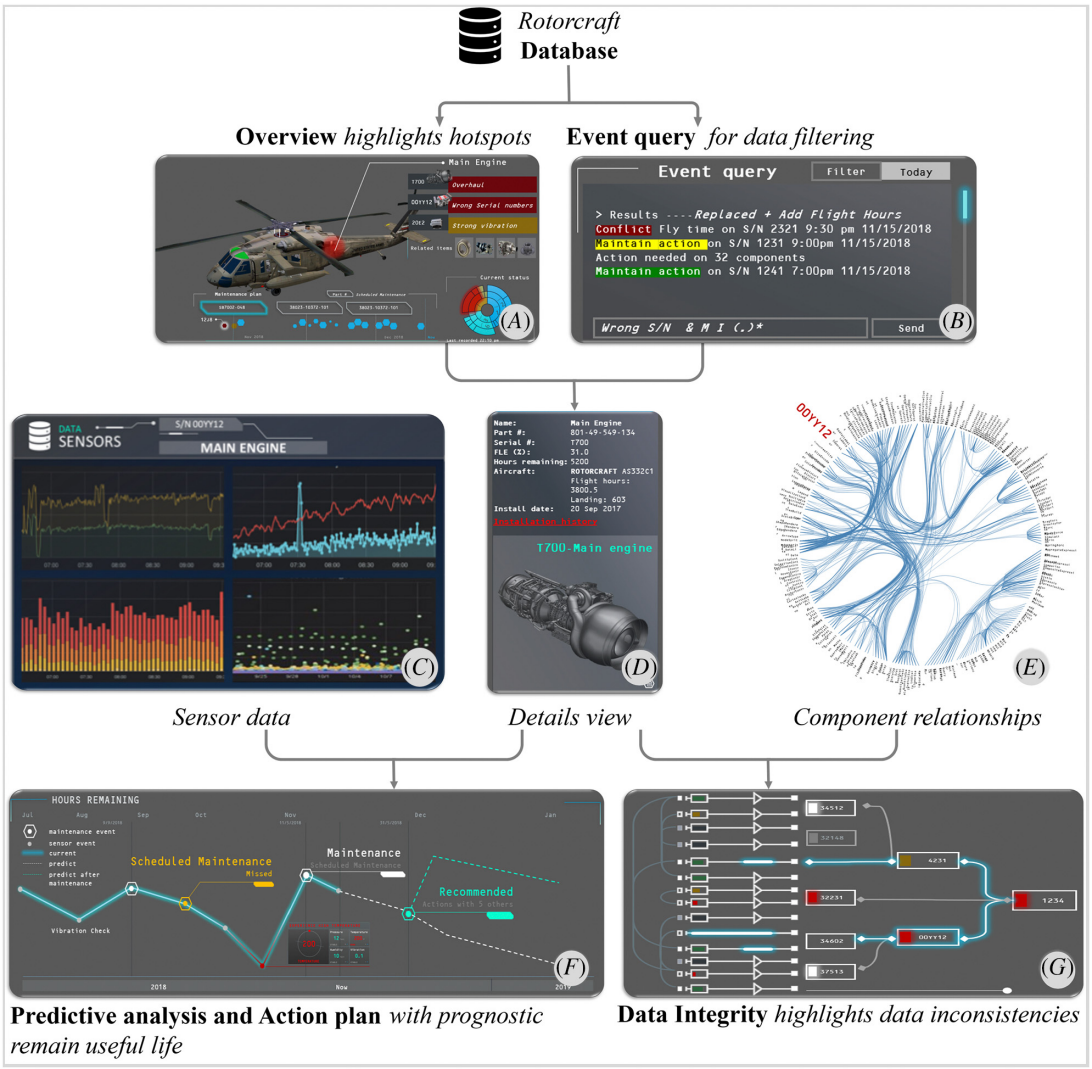
The process of creating visualization dashboards involves several steps, shown in **figure 2**. Data are retrieved through Web Application Program Interface, known as Web API services, and then transcribed into the scene by using Three.js,[28] a cross-browser, which uses Web Graphics Library, known as WebGL. JavaScript library/API is used to create and display animated three-dimensional (3-D) computer graphics in a web browser. Dashboards will be integrated into the interface using D3.js.[29] Since visual interfaces are developed in the web environment, they can be naturally deployed on aircraft notebooks or smart devices/tablets with a web browser. In addition, our visualizations can link and display the outputs from the data mining component, which is also implemented in JavaScript using tensorflow.js[30] released by Google.

The visualization dashboard presented in **figure 2** supports the following design goals:

a) Enable automated component tracking capability, which will allow engine records and health state awareness information obtained from each aircraft serial number (S/N) to be autonomously compared to the data from the fleet so aviation stakeholders can track the current trend at local (e.g., theaters, aviation depot sites) as well as global (e.g., logistics, acquisition echelons) levels in real time or near real time. If anomalies or irregularities, whether they are related to data integrity, security, or component fault, are found, visual flags and colored alerts will be triggered for the aviation fleet health state awareness.[31]

b) Allow maintainers or engineers to scan the Radio-Frequency Identification chips or other onboard data acquisition devices, including HUMS, to pull data such as rotorcraft component records, including aircraft configuration, flight hours, usages, and remaining useful life (RUL). It can also leave digital notes to share information with other users or inform certain critical maintenance information to appropriate organizations or links within the Global Combat Support System-Army echelon to take required actions.

FIG. 2 Schematic overview of visualization framework: (*A*) overview of hot spots, (*B*) event query for data filtering, (*C*) sensor data monitoring, (*D*) detail view of the selected component, (*E*) component relationship, (*F*) predictive analysis and action plan, and (*G*) data integrity graph.



c) Detect data inconsistencies or irregularities, inform discovered issues, and autonomously enable the data refinement.

Specific capabilities supported by visualization techniques include the following:

a) Detection of outliers: The system can automatically highlight where the hot spots[32] are, as shown in **figure 2*A***, and identify problems proactively, improving both reliability and safety. The types of outliers, which can be captured in our system, include time series abnormalities[33] (e.g., sudden increase or decrease in engine temperature[31]), multidimensional outliers that might not be detectable in the marginal distributions,[34] and both (e.g., multidimensional time series outliers).

b) Visual characterizations: In addition to detecting outliers, the system can programmatically characterize other visual features of high-dimensional data, such as density, skewness, shape, and texture.[35,36] Working directly with these visual characterizations, users can quickly narrow down interesting subseries or unusual

correlations between variables within high-dimensional datasets.[12] This capability is especially useful as our system and interfaces are designed to handle real-time streaming data.

c) Proactive and predictive analysis: The developed techniques can project potential threats (e.g., poor engine performance) and predict the time it takes to have adverse effects (e.g., degraded efficiency) or detect malicious attacks (e.g., engine anomalies/irregularities or faults/degradations) and determine the time it takes for the system to fail (e.g., engine failure). The models are trained on historical data, make real-time or near-real-time predictions, and raise the alarm to the users, including pilots, for timely actions. The machine-learning framework can also use data fused from other sources such as sensor data, including vibration, temperature, pressure, stress and virtual loads, and humidity, as shown in **figure 2C**. Component health status and sensor data can serve as inputs for our threats or cyberattack detection mechanisms.

d) Component relationships and integrity assurance: As depicted in **figure 2G**, the visual interface can display component interconnections[37] and potential impacts on other subsystems when the integrity of one component is compromised.

In general, the visualization techniques and prototype being developed can be used to assist the users to track possible threats (e.g., engine's poor performance) or malicious behaviors (e.g., engine anomalies/irregularities or faults/degradations) to project the health state of the rotorcraft gas turbine engines and their components and proactively advise the stakeholders of the potential issues, RUL, and potential mitigation actions. The developed prototype is designed to support intuitive and interactive features.[38]

The cybersecurity dashboard can support a full range of visualization spectrum from basic charts (e.g., line graphs) to multidimensional plots (e.g., radar charts). As the system autonomously performs the data integrity check across the relational database of high-dimensional data, missing information, errors, and abnormal inputs can be captured to inform the users to verify or address the discovered issues. Once data are populated to the system, users can visually identify where the hot spots are and their related components that need attention. The visualization system, as shown in **figure 3**, will provide the visualization solutions integrated with data analytics, highlight the patterns, and trigger component anomalies. Users can attain the granularity of the presented information via the user-friendly interactive features, explore the actions to be performed, and then digitally update the records with new data once completed. Our visualization system adopts the three-step interactive framework[39]:

a) Overview of the rotorcraft fleet and individual aircraft S/N: The main views aggregate a large amount of data to highlight patterns and trends from the fleet and individual aircraft. Issues from aircraft will be highlighted via color codes by urgency priority (e.g., imminent threats). As depicted in **figure 3**, a list of S/Ns of helicopters and the corresponding time series thumbnails are presented for the initial exploration.[40]

b) Zoom and filter: Users can touch the aircraft S/N to project the 3-D aircraft model with components highlighted by color codes depending on, for example, its health state or maintenance status. Additionally, users can filter the aircraft activity patterns using the event query in forms of both textual sequence and graphical input.[41]

c) Details on demand: With simple queries or mouse clicks, technical details (e.g., multivariate time series charts) can be visually presented to provide viable knowledge of the selected component. In **figure 3**, for example, a 3-D helicopter model shows a potential threat in its tail rotor and engine via red-blinking highlight. When the user clicks on that color-changed component, e.g., engine, it would show the multivariate heat-maps of that part (e.g., historical versus real-time temperature readings, flight time, maintenance schedule, and RUL) or the characteristics of discovered issues related to the information integrity, as depicted in the right screen panel of **figure 4**.

Besides transforming the current data into meaningful and actionable information, the system can also proactively learn from the historical records to formulate a comprehensive solution that users or decision-makers can fully understand in order to assess individual rotorcraft and fleet readiness accurately and effectively. The

FIG. 3 Our visualization dashboard of the rotorcraft database.

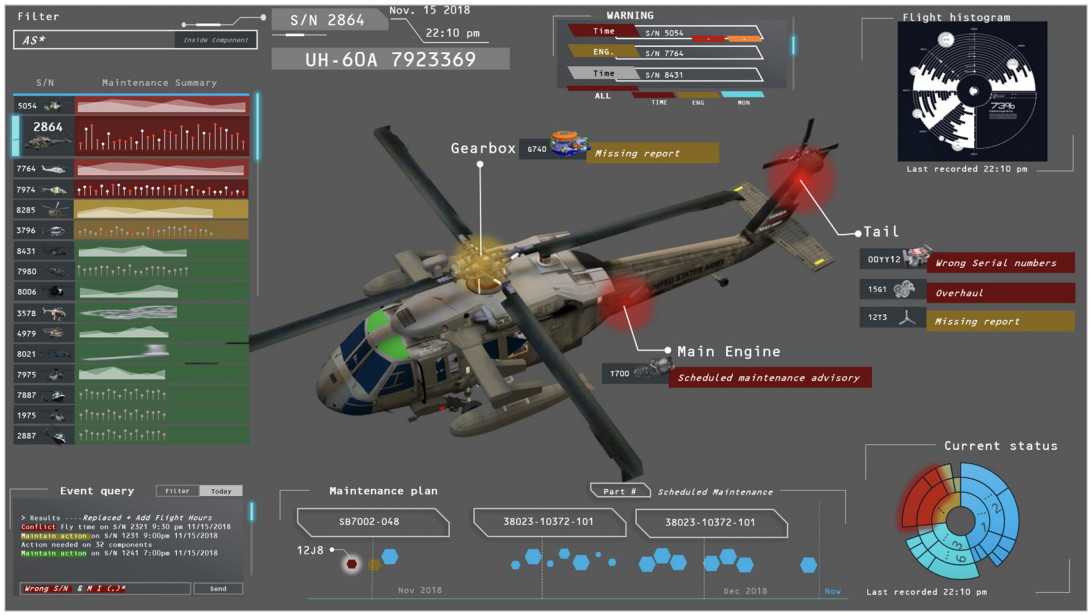


FIG. 4 Interactive operations (e.g., pan and zoom, details on demand) supported by our conceptual analytics framework.

graphical front-end feature is designed to capture and present the back-end predictive analytics on sensor data and flag potential security threats, which will be discussed in the next section.

PREDICTIVE ANALYTICS FOR AIRCRAFT THREAT MONITORING AND MITIGATION ACTIONS

The IoT devices in an industrial manufacturing environment are the sources of a large amount of heterogeneous real-time or near-real-time data, which have the potential to provide valuable information in manufacturing processes. The promising outcomes could be utilized for prognostics, security detection, and smart manufacturing

control. However, analyzing these heterogeneous and noise-prone data from these ubiquitous devices to extract useful information is a huge challenge. Therefore, an IoT-based machine learning solution, which comes from analyzing a vast amount of industrial data and supporting online monitoring and smart manufacturing control, can become critical. This solution is a combination of different technologies, such as big data analytics, machine learning, and artificial intelligence for industrial operations.

This section discusses experimental results from a case study of a data-driven machine learning approach to analyze industrial IoT data collected from sensors used to monitor the health condition of aircraft engines. This use case is to demonstrate that the machine learning back-end feature of our solution can extract useful information from the noise-prone data collected from ubiquitous factory devices. There are two main approaches to aircraft health status prognostics: physics-based and data-driven approaches. The latter is gaining favor because of the advancement of the machine learning field (e.g., mainly Neural Networks) and the publication of four run-to-failure datasets of a turbofan engine simulation model obtained from NASA's PCoE in 2008.[15] The simulation model is Commercial Modular Aero-Propulsion System Simulation (C-MAPSS).[42]

Table 1 shows a typical data format for the four C-MAPSS turbofan engine datasets, containing 27 attributes of numerical values. Each row is a record of an operational cycle of an individual engine unit. There are three operational settings, called Altitude, Mach Number, and Throttle Resolver Angle. These parameters have a direct effect on engine performance. The next 21 fields indicate sensor data collected from Sensor 1 to Sensor 21, which have been recorded from each cycle and are contaminated with sensor noise. There are a training set and testing set for each data set. In the training set, the engine ran to failure. In the test set, the time series ended sometime before the engine fully degraded or failed, and the prognostics algorithm was tasked to predict the RUL.

This case study uses these four datasets because their characteristics resemble sensor data collected from the IoT-based manufacturing automation environment. These datasets are the result of the incorporation of different fault injection parameters to simulate degradation trends or anomalies. Sensors were strategically placed in various parts of the system to collect characteristics of the engine over the operational time. Two main critical aspects of these datasets influence our decision to use them in this case study. First, the datasets represent temporal, multidimensional, and nonlinear sensor data collected from a simulation model of a complex industrial system. Second, noise was incorporated to describe the nature of variability in the industrial IoT-based manufacturing automation environment.

After the publication of these four datasets, many researchers around the world used them to formulate and test their prognostics algorithms. However, there was no common evaluation standard among these publications, leading to difficulty in comparing their performance. In 2014, Ramasso and Saxena[43] did a thorough survey of about 70 publications that used one or more of these four datasets. This survey also provided a benchmark for other researchers to develop and compare their algorithms. In this article, the authors also pointed out the three winning methods, which were (1) Similarity-Based Approach,[44] (2) Recurrent Neural Network Approach,[45] and

TABLE 1
Typical format of turbofan engine datasets

Engine	Cycle	Altitude	Mach #	TRA	Sensor 1	…	Sensor 21	RUL
1	1	34.99	0.84	100	449.44	…	8.80	148
1	2	41.98	0.82	100	445.00	…	6.26	147
…	…	…	…	…	…	…	…	
1	149	42.01	0.84	100	445.00	…	6.22	0
2	1	34.99	0.83	100	449.41		6.26	130
2	2	34.98	0.81	100	446.33		6.26	129
…	…							

Note: TRA = Throttle Resolver Angle. The three dots notation (…) indicates more data are available.

(3) Multilayer Perceptron and Kalman Filter Based Approach.[46] Since then, with the maturation of neural network methods such as Long Short-Term Memory (LSTM) Neural Networks[47] and Convolution Neural Networks (CNN),[48] notably in Zheng et al. in 2017[49] and Jayasinghe et al. in 2018[50] the authors made use of LSTM and produced the best results in terms of mean squared error (MSE) for datasets 1 and 3. The latter combined LSTM and CNN to achieve better MSE results for Datasets 2 and 4.

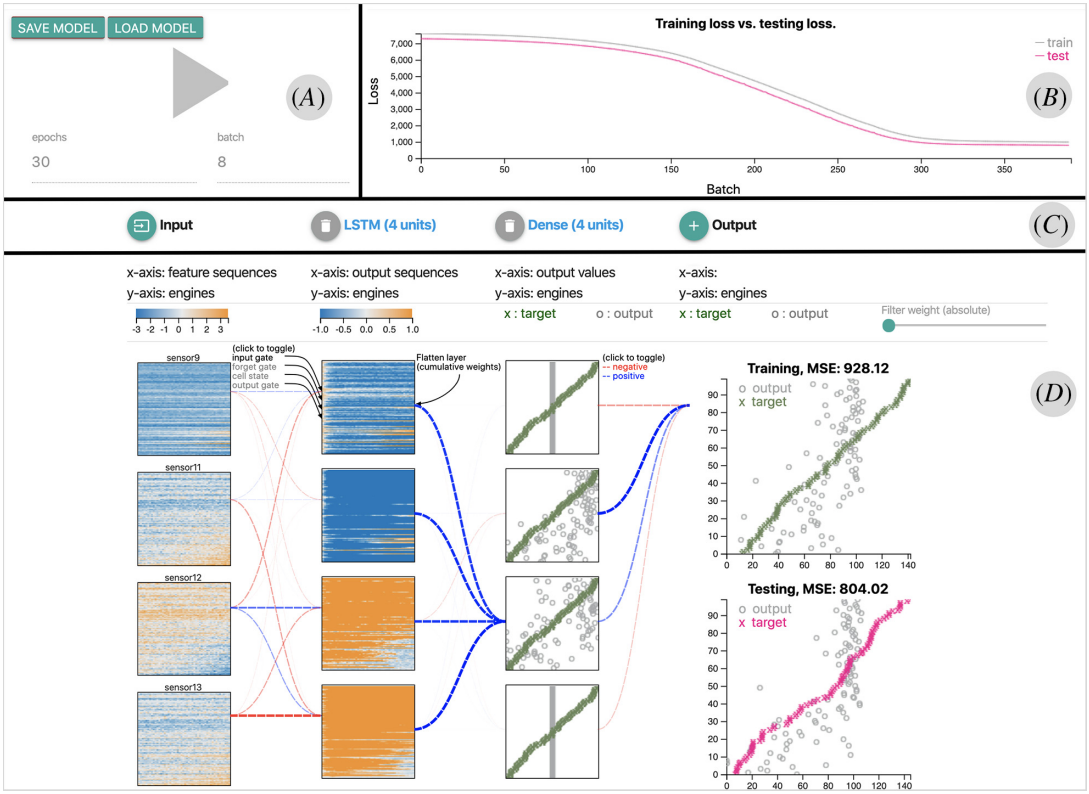
For all the publications surveyed in 2014, there was no standard benchmark among these publications, leading to the difficulty of suitably comparing their results. This section discusses two recently published works with noticeable results. There is no source code available for the LSTM approach proposed by Zheng et al. in 2017.[49] Also, several implementation aspects are obscure in this publication, leading to difficulty in reconstructing the work. The first one is about the training data generation process. Merely using the provided sensor data would lead to RUL values of zero for all the training units, meaning that the neural network would not be able to learn. Therefore, there must be a training data generation method to be used, but this information is not introduced in the cited article. Similarly, there were several model configurations (e.g., hyper-parameters) used in the solution that were not clearly defined. For instance, the authors specified that they used dropout layers but did not mention where the dropout layers were in the neural network and what dropout rates were used. The authors also specified that they used regularizers to reduce overfitting but did not tell what type of regularizers (such as Lasso Regularization, known as L1, or Ridge Regularization, known as L2, or others) or regularization rates were used, or to which layers they applied the regularizations. On the other hand, the work proposed by Jayasinghe et al. in 2018[50] has the source code available for implementation, but the time it takes to train the model is long. Long training time leads to difficulty in adapting it in an industrial IoT-based environment. Also, with the noise-prone data collected from this aircraft engine complex system and the stochastic nature involved in the training process of Neural Networks, there should be more testing to confirm the stability of the results specified.

There are two main principles for our predictive framework for a machine learning back-end feature. They are human-in-the-loop machine learning and customizability. All the publications specified previously are purely data-driven approaches. These approaches cannot exploit the human domain of knowledge, especially in the complex system with noises as an unavoidable part of data collected from the industrial IoT-based environment. Our approach is to combine the data-driven approach with human knowledge. Several models are developed that are based on the two previously mentioned publications with the current state-of-the-art methods. We have also incorporated visualization solutions for these models. These solutions allow the users to visually view the input, final output, and even intermediate outputs from the middle layers of the trained models. Therefore, the machine learning models are no longer a black-box to the users. Using domain knowledge, cognitive capability, and visual representations of intermediate steps (such as weights and intermediate outputs of each layer), users can understand the prediction process. Users could follow why the models made that prediction and what input from one layer to the next layer the models used to make the prediction and determine if the forecast was more accurate. Therefore, users can decide to select a result from a single model or even combine multiple results from different models.

As in this complex environment, no single model may work best for all the datasets. For instance, the model proposed by Zheng et al. currently works best for Datasets 1 and 3, while the one proposed by Jayasinghe et al. is the best for Datasets 2 and 4. Also, users may have different requirements for models, such as run time, power efficiency, and computing power requirements. Therefore, our approach provides users with the ability to inject their knowledge into the learning process. Users can do this via customizable model hyper-parameters such as number of layers, number of hidden units, weights in each layer, regularizer types, and regularizer parameters.

There are several components supporting visualization and customizability of the machine-learning process. **Figure 5** shows an overview of our machine learning solution, including four panels (*A*)–(*D*). Panel (*A*) consists of controller and general parameters for the system. Panel (*B*) provides a training and testing loss graph. Panel (*C*) includes input, hidden layers, and output configurations, and panel (*D*) presents detailed architecture of the neural network inside the framework.

FIG. 5 Machine learning technique overview: (*A*) controller and general parameters; (*B*) training and testing loss graph; (*C*) input, hidden layers, and output configurations; and (*D*) detailed architecture of the neural network.



We begin with panel (*A*) (see **fig. 5*A***). The framework allows users to configure the Neural Networks, such as selecting the number of epochs and batch size. Panel (*B*) (see **fig. 5*B***) shows the training and testing loss graph of the current configuration. The higher number of epochs leads to lower training loss but may lead to higher testing loss due to overfitting. In this case, users should stop the training process after fewer epochs. Users also have options to set the batch size and number of epochs before training from the options below the start/pause training button.

Panel (*C*) (see **fig. 5*C***) consists of options for input, hidden layer, and output configurations and has a close-up view in **figure 6**. Our solution implements a default neural network with the main configurations suggested in Zheng et al.[49] Besides, it also provides models with the number of hidden units reduced for faster training time and comparison. A layer can be completely removed from the network via a delete button (see **fig. 6*A***). When the option of adding layers is selected (see **fig. 6*B***), a pop-up menu (see **fig. 6*C***) appears with configuration information of layer type, the number of units the layer contains, and an activation function. Adding a layer may produce better predictions at the cost of training and prediction time. However, complicated models (e.g., with more hidden layers) might result in overfitting. In other words, it fits too well to the training data but is not generalized enough to give good predictions on the testing data. Besides modifying the layer at large, this solution also allows users to change the number of hidden units in each layer.

Figure 7 presents the neural network architecture. After loading, the system displays the input in the form of heat-maps (see **figs. 7*A*** and **8*A*** for a close-up view), e.g., one heat-map per sensor input. The x-axis of each heat-map is the sensor data sequence (e.g., the number of cycles operated), while its y-axis is the name of engine units (e.g., 1-100 for Data Set 1). The heat-map intensity at a specific point represents the sensor's value of a

FIG. 6 Layer configurations: (*A*) delete button, (*B*) adding layer button, and (*C*) pop-up menu.

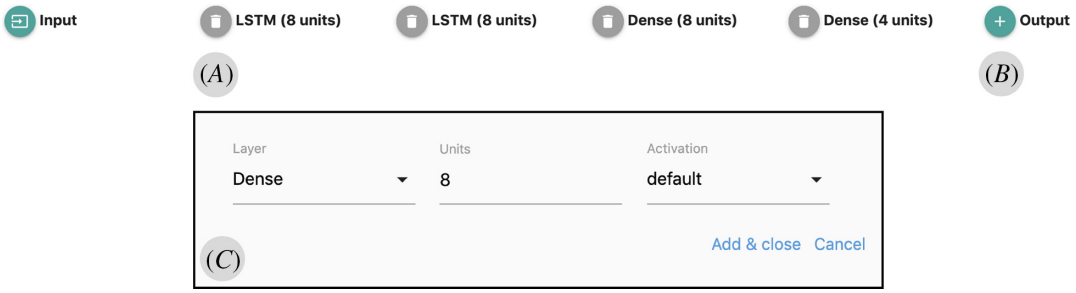
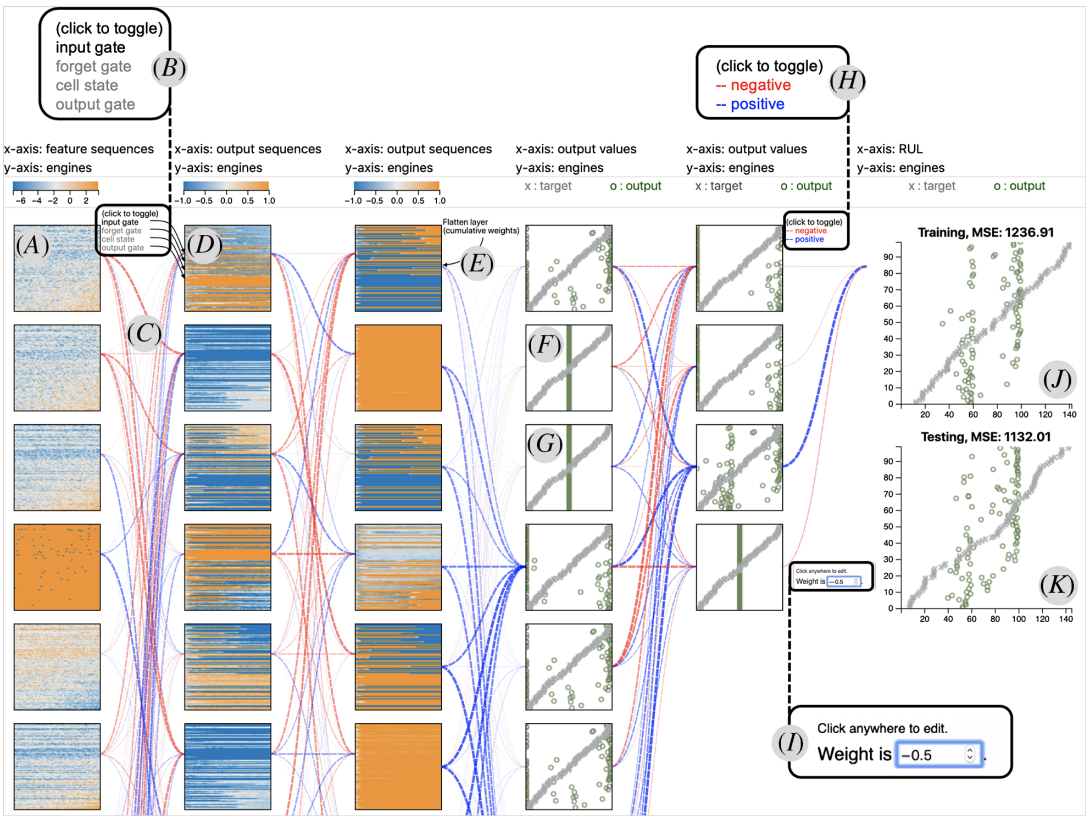


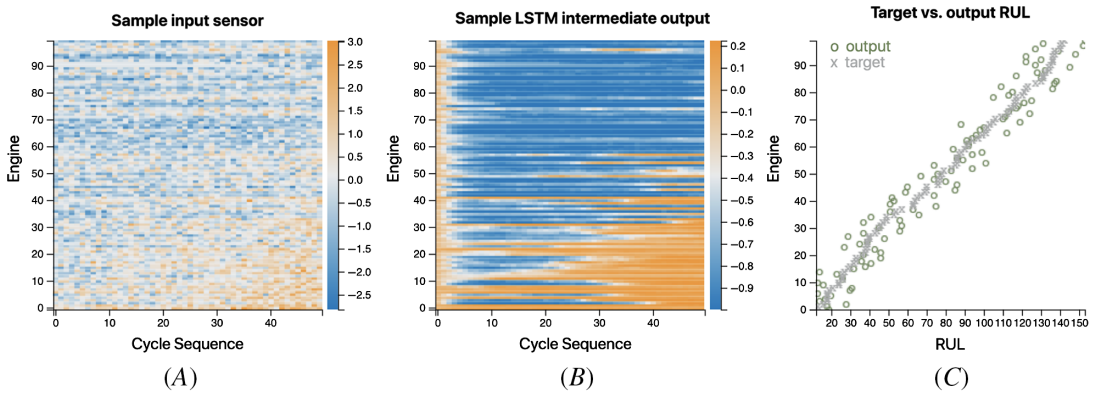
FIG. 7 Neural Network architecture. (*A*) Input in the form of heat-maps; (*B*) and (*H*) close-up views of the menu to toggle displays; (*C*) network weights of each layer; (*D*) output from LSTM layer; (*E*) result of flattening from an LSTM layer; (*F*) and (*G*) outputs from Dense layer; (*I*) weights tuning; (*J*) and (*K*) outputs of the training and testing data.



corresponding engine at a time step accordingly. The input units are also sorted by their RUL to reveal the time-dependent patterns. After setting up the number of hidden layers and corresponding hidden units, users can click on the training button to start the learning process.

Similar to the input layer, the outputs from LSTM layers are also sequences with the corresponding number of engines and steps. The system also represents them as a heat-map (see **figs. 7*D*** and **8*B*** for a close-up view) with a similar specification to be coherent. The Dense layers and final training/testing outputs are visualized as scatter

FIG. 8 Close-up views: (*A*) input in the form of heat-maps, (*B*) output from LSTM layers, and (*C*) output of the training data.



plots. The y-axis also represents the number of engines, and the x-axis describes the output values. The circles at the scatter plot represent the predicted outputs of the corresponding engine units as inputs. The *x* symbols are the actual target RULs. The two are linearly scaled to the domain of target RULs for better visualization. The outputs of the training data (see **figs. 7*J*** and **8*C*** for an example close-up view) and testing data (see **fig. 7*K***) at the last layer of the visualization have larger sizes and visible axes to provide unobstructed views for the users to see the final training and testing results.

As shown in **figure 7*F*** and **7*G***, outputs from Neurons 2 and 3 of Dense Layer 2 are similar, with all data points producing a constant value, and hence do not contribute to the training process. Therefore, users could try to reduce one from the number of units for this layer (using the model configuration menu) and retrain the model to gain training/predicting time.

Users can observe the network weights of each layer (see **fig. 7*C***) to see how each feature contributes to the output in the next layer. The line thickness represents the network weights. The thicker the lines, the higher the weight values. The number of weights may get high in several cases. For instance, if more hidden nodes were used, LSTM weights might have different types, or weights could come from a flattened layer. If there were many weights to visualize, it would lead to slow rendering and cluttering of lines. There are two approaches to overcome these issues. One is filtering by weight types, and another one is accumulating weights from a single feature into one for displaying purpose. For the first approach, the system allows users to filter out weight lines by LSTM weight types (e.g., input gate, forget gate, cell state, and output gate). **Figure 7*B*** is a close-up view of the menu to toggle displays of LSTM weight types. Users can click on "click to toggle" to toggle all LSTM weight types or click on each type to toggle the corresponding weights. Similarly, **figure 7*H*** shows the close-up view of the menu to toggle displays of either positive or negative weights.

After flattening from an LSTM layer (see **fig. 7*E***), there will be many weights to pass the sequence feature to the next Dense layer. For instance, 50 steps in a feature sequence flattened from an LSTM layer with eight nodes to the next Dense layer with four hidden units will create 50 x 8 x 4 = 1,600 weights to display. Our second approach is to deal with this case. The system accumulates the steps of a sequence feature into one weight for a display purpose. The reason is that they all represent the contribution of one feature from the LSTM to the next layer. In the previous example, this reduces to 8 x 4 = 32 paths to display. Also, this gives a clearer and better overview of the contribution of each LSTM feature.

With subject matter knowledge and experiences, users understand the significance of a feature in the results. Therefore, they might wish to tune the component/link weights in the neural network as depicted in **figure 7*I***. This functionality is essential since the learning process is stochastic, and in several cases, it would not converge or could converge into local minima. This human-in-the-loop machine-learning feature allows

TABLE 2
Different neural network configurations experimented

	Input Features	LSTM1	LSTM2	Dense1	Dense2
Configuration 1	15	64	64	8	8
Configuration 2	15	8	8	8	4
Configuration 3	8	8	8	8	4
Configuration 4	7	8	8	8	4

the users to inject their domain knowledge, cognitive capability, and understanding of the visual representations into the learning process.

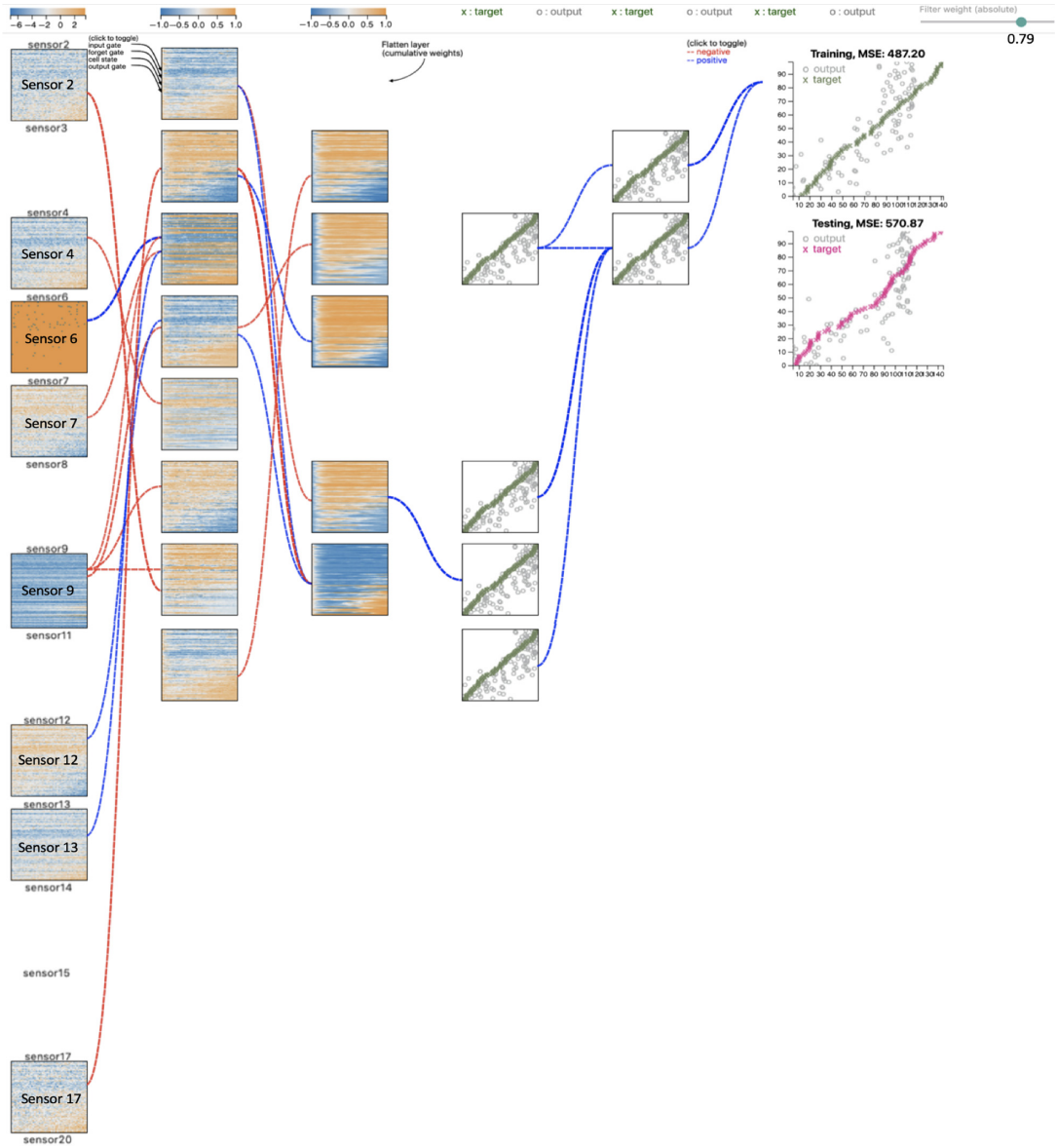
Table 2 shows four different neural network architectures experimented with using our framework to build models for predicting RUL from the first data set (e.g., out of the four C-MAPSS datasets). Configuration 1 is a neural network suggested in Jayasinghe et al.[50] Specifically, this configuration has two LSTM layers, in which each has 64 hidden units, and two Dense layers, in which they have eight hidden units, correspondingly. On the other hand, Configuration 2 is a simpler version of this architecture, with numbers of hidden units reduced for faster training time. This reduction of training time allows us to save time while exploring other configurations for the contributions of different sensors on the final prediction results. Both Configurations 1 and 2 use all 15 sensors (2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, 21) suggested in Wang et al.[44] However, explorations, using the weight filtering functionality as described earlier, suggest that there are eight sensors (2, 4, 6, 7, 9, 12, 13, 17) that are more significant than the others (3, 8, 11, 14, 15, 20, 21). Figure 9 is an example of such an exploration. It is the view for the 7th model of Configuration 2 (see Table 3) with the weight filter set to 0.79. The eight significant features are still having weights connected to them while the others are gone since there are no significant weights attached to them. Therefore, Configurations 3 and 4 are used to test the accuracies of the models using these 8 and 7 sensors correspondingly.

Table 3 shows the experimental results for the four mentioned neural network configurations. All executions are on an iMac desktop computer with operating system macOS Mojave (Version 10.14.6), 3 GHz Intel Core i5, and 8 Gygabyte Memory. Each setting was executed ten times, and then the training times (in milliseconds) and the MSEs together with their means and standard deviations are reported correspondingly. It is worth noticing that the MSE values in red are the outlying MSEs since their corresponding models are locked in their local minimums during the learning process. Therefore, we remove these values while calculating the confidence interval for the difference between the two means of the compared quantities (discussed later) using the t-test.

It is observable that execution times are reasonably stable while the MSEs have high variances due to the stochastic nature of the neural network training process. Configuration 2 (after reducing the number of hidden units) is significantly faster to train compared to Configuration 1. The faster training time comes with a cost of accuracy. The 95 % confidence interval for the difference between the two means of these two values of MSEs is (−257.4229, −113.3071). This confidence interval means that the average of the MSEs of the models in Configuration 2 is significantly higher than that of Configuration 1. Training time versus accuracy is the trade-off that the system analysts often need to do when training neural network models. In this case, Configurations 3 and 4 are using simpler settings to speed up our experiments about the contributions of individual sensors to the final prediction results.

Configuration 3 (with 8 significant input features) and Configuration 4 (with 7 less significant input features) are not much different in terms of training time compared to Configuration 2 (with all 15 input features). These less significant differences mean that, in this case, reducing the number of input features does not help in improving training time. Regarding MSEs, the 95 % confidence intervals for the differences between the two MSE means of Configuration 2 versus Configuration 3 and Configuration 2 versus Configuration 4 are (−72.4582, 25.6682) and (−112.4778, −39.5842), correspondingly. The first confidence interval means that reducing from all 15 input features to 8 important features does not affect the accuracy substantially. However, the latter shows

FIG. 9 Exploring significant features using weight filtering functionality.



that using seven less significant sensors for training models harms the performance significantly. These differences prove the fact that system analysts can use our proposed framework to select important input features without cutting the prediction accuracy significantly. The ability to select relevant features is significant in several situations, such as the need to reduce the number of sensors to save data storage and hardware costs in manufacturing.

For the reproducibility of these results, interested readers can refer to the web page of our prototype at https://idatavisualizationlab.github.io/V/RUL_Viz/.[51] Using this web page, users can load the models trained and stored on the server for further investigations. These results and discussions show that it is relatively simple for users to re-implement a complicated solution, such as the one published in Zheng et al.,[49] for the C-MAPSS data set using the proposed framework. Furthermore, our solution allows users to explore what is inside the model via the visualizations of the intermediate, extracted features, and corresponding weights. These details give the

TABLE 3
Training time (*ms*) and MSE for different neural network configurations

	Configuration 1		Configuration 2		Configuration 3		Configuration 4	
No	Time	MSE	Time	MSE	Time	MSE	Time	MSE
1	1,131,649	383.91	419,696	667.84	479,975	708.69	438,804	716.96
2	1,141,658	458.12	485,000	664.18	471,303	590.64	425,672	746.56
3	1,140,010	505.02	471,856	658.18	478,691	730.85	426,540	637.63
4	1,107,596	320.76	498,024	669.04	490,019	724.36	427,614	*1,118.03*
5	1,140,451	*1,464.63*	495,276	705.53	491,262	641.25	425,239	716.39
6	1,175,071	*1,210.44*	497,241	651.91	491,785	734.05	424,274	760.66
7	1,171,033	412.76	496,606	570.87	492,621	653.56	425,942	721.50
8	1,123,494	504.13	494,059	644.06	491,342	*7,322.97*	423,615	725.66
9	1,098,431	597.65	492,582	614.16	489,704	614.16	424,304	759.15
10	1,104,206	545.13	490,692	667.23	488,812	*1,046.92*	424,373	761.47
Mean	1,133,360	465.94	484,103	651.30	486,551	674.70	426,638	727.33
StDev	26,154	90.47	23,947	36.36	7,257	56.81	4,443	38.68

Note: Execution number (No); Standard deviation (StDev); the MSEs in italics are outlying.

users a thorough understanding of the built models. This understanding also provides directions for users to explore other configurations. Thus, the proposed framework offers the required functionalities to support users in exploring new model configurations.

Levels of granularity for detail have been incorporated in visual representation to support users in getting a good grasp of the architecture of the network. Each line connecting any two layers represents the value of the corresponding weight. The system provides a filtering feature to unclutter the overall view and keep the desired type of entries. Particularly, users can filter out the weights and keep the desirable ones by utilizing the toggle menu by which the weights in selected types remain on display. The underlying model stays the same, but the visual representation highlights the significant weights by selection.

The proposed framework introduces intelligent visual analytics, leverages the concept of human-in-the-loop in machine learning, and employs them to secure a manufacturing automation system. By enabling users to configure the model structure intentionally, each network can be considered as a component in the process of determining the most proper model for use. The system offers the training and testing results as well as the corresponding visualization for those results. From the visual presentation result and the comparison between configuration alternatives, users can identify critical features, e.g., what the common features of adequate configurations are, so they can focus on them. This procedure can be considered as an act of opening the "black-box" of such a neural network model. State-of-the-art Neural Networks can sometimes be fooled and then misclassify using adversarial examples, e.g., the ones that are intentionally modified to cause perturbations from correct classification examples.[52] The internal structure of such a neural network is explicitly explainable, which enables users to determine where in the process such a misclassification originates. The progress of testing a new set of parameters is not random but based on previous outcomes and understanding of the characteristic of the features. By removing the insignificant nodes in the model structure, we can also reduce the load on time and resources such as memory and computing power.

The presented case study adopts the machine learning process for prognostics of aircraft turbofan engine RUL. However, as previously discussed, we have generalized these datasets to simulate the sensor data collected from typical industrial IoT-based settings. Therefore, our framework is transferable to other use cases, such as learning the abnormal patterns of health status data, sensor data, or power usage. These may be indications that a component in the manufacturing process is under attack/breach. Abnormal power usage or data transfer rates might serve as an indicator of cyberattack on an industrial IoT device. Specifically, instead of the RUL column in **Table 1**, users could collect data about the threat or attack information, such as attack time, attack type, and RUL, after an attack occurred. In this case, our technique could learn from the future collected sensor data about potential attack time,

attack models, or changes in RUL. These behaviors could be learned, predicted, and visualized on the system dashboards described earlier. The system can then accordingly recommend timely actions.

Discussion of Research Potential Benefits

The framework, as shown in **figure 1**, and developing analytics and data visualization and mining techniques are designed for securing efficient manufacturing and can be used for Industry 4.0 to build a resilient and efficient manufacturing automation system. With humans in the loop at the command center, intelligent solutions are autonomously formulated and incorporated with influential manufacturing parameters (e.g., efficiency, peak load, fuel consumption, setup time, recovery time, minimum energy loss, meantime-to-detect, and other operational parameters, or both), which have a direct relationship between, for example, maximum and minimum efficiency loss. Manufacturing efficiency metrics, however, need to be frequently updated by the stakeholders and operators to reflect the existing risks, adverse conditions, and capability of state-of-the-art artificial intelligence and machine-learning techniques and tools as well as other related monitoring technologies.

In the case of aircraft component automated tracking and aircraft health state awareness technologies, the developing analytics and data visualization and mining techniques, when incorporated into the holistic framework illustrated in **figure 1**, can not only detect security threats and attacks but also autonomously direct how the aircraft may fly, via the reconfigurable flight controls technology, to avoid severe usages, which in turn can minimize the specific fuel consumption. With the growing demands from air traveling, fuel consumption efficiency and savings can be extremely valuable.

Conclusions

This article presented a visualization technique that aims to improve security while improving the efficiency of complex systems or networks. The proposed data analytics and mining techniques were tested and validated using aircraft engine datasets to simulate the ability to project potential threats (e.g., poor engine performance) and predict the time it takes to have adverse effects (e.g., degraded efficiency) or detect malicious attacks (e.g., engine anomalies/irregularities or faults/degradations) and determine the time it takes for the system to fail (e.g., engine failure). When we can obtain more adequate datasets on manufacturing automation networks, we plan to use them to validate the proposed framework and techniques for securing the efficient manufacturing automation process.

The proposed system provides functionalities with levels of granularity corresponding with the skill level of the end-user. The first level with general analysis requires an insignificant cognitive load as well as specialized knowledge. For instance, the model of the neural network demands adequate understanding of machine learning and Neural Networks in general. A higher level of features supports complex functionalities with in-depth analysis and monitoring.

Regarding scalability, our proposed system is scalable to work with more models, users, and data. For the current implementation, the client browser executes both the back-end model training and front-end visualization. Therefore, individual users can work with developed models at their computers, and thus there is no issue for serving more users. Also, the save/load model to/from files functionality allows analysts to train different models and share them easily. On the other hand, developers can convert the current system codes (e.g., in JavaScript) to a server-based application (e.g., using the Node.js server). The server-based deployment allows training models that need to learn from a large amount of data. Specifically, the server acts as the back-end model training, and then the trained model can be displayed and explored at the client browser.

ACKNOWLEDGMENTS

The authors greatly appreciate the TTU Office of Research and Innovation, Edward E. Whitacre, Jr. College of Engineering, and Computer Science Department for valuable support and guidance in research efforts related to the manufacturing cybersecurity and efficiency. The authors would like to thank the ASTM reviewers for their insightful comments that significantly improved the article and acknowledge the inclusion of some of their ideas.

References

1. Verizon Enterprise, "2017 Data Breach Investigations Report," Verizon Enterprise, 2017. http://web.archive.org/web/20191030003917/https://www.ictsecuritymagazine.com/wp-content/uploads/2017-Data-Breach-Investigations-Report.pdf
2. Department of Energy, "DOE Announces $70 Million for Cybersecurity Institute for Energy Efficient Manufacturing," Department of Energy, 2019. http://web.archive.org/web/20191030035111/https://www.energy.gov/articles/doe-announces-70-millioncybersecurity-institute-energy-efficient-manufacturing
3. S. R. Chhetri, S. Faezi, N. Rashid, and M. A. Al Faruque, "Manufacturing Supply Chain and Product Lifecycle Security in the Era of Industry 4.0," *Journal of Hardware and Systems Security* 2, no. 1 (March 2018): 51–68. https://doi.org/10.1007/s41635-017-0031-0
4. S. Mansfield-Devine, "Securing Small and Medium-Size Businesses," *Network Security* 2016, no. 7 (July 2016): 14–20. https://doi.org/10.1016/S1353-4858(16)30070-8
5. E. Osborn, *Business versus Technology: Sources of the Perceived Lack of Cyber Security in SMEs, CDT Technical Paper 01/15* (Oxford, UK: University of Oxford, 2014).
6. Y. Wang, O. Anokhin, and R. Anderl, "Concept and Use Case Driven Approach for Mapping IT Security Requirements on System Assets and Processes in Industrie 4.0," *Procedia CIRP* 63 (2017): 207–212. https://doi.org/10.1016/j.procir.2017.03.142
7. N. Tuptuk and S. Hailes, "Security of Smart Manufacturing Systems," *Journal of Manufacturing Systems* 47 (April 2018): 93–106. https://doi.org/10.1016/j.jmsy.2018.04.007
8. T. C. Mahoney and J. Davis, *Cybersecurity for Manufacturers: Securing the Digitized and Connected Factory, Report Number MF-TR-2017-0202* (Ann Arbor, MI: MForesight, 2017).
9. J. Mehnen, H. He, S. Tedeschi, and N. Tapoglou, "Practical Security Aspects of the Internet of Things," in *Cybersecurity for Industry 4.0* (Cham, Switzerland: Springer, 2017), 225–242.
10. T. Spyridopoulos, T. Tryfonas, and J. May, "Incident Analysis & Digital Forensics in SCADA and Industrial Control Systems," in *Eighth IET International System Safety Conference Incorporating the Cyber Security Conference* (Stevenage, UK: Institution of Engineering and Technology, 2013), 1–6.
11. L. Thames and D. Schaefer, "Cybersecurity for Industry 4.0 and Advanced Manufacturing Environments with Ensemble Intelligence," in *Cybersecurity for Industry 4.0* (Cham, Switzerland: Springer, 2017), 243–265.
12. D. Glavach, J. LaSalle-DeSantis, and S. Zimmerman, "Applying and Assessing Cybersecurity Controls for Direct Digital Manufacturing (DDM) Systems," in *Cybersecurity for Industry 4.0* (Cham, Switzerland: Springer, 2017), 173–194.
13. T. N. Dang, A. Anand, and L. Wilkinson, "Timeseer: Scagnostics for High-Dimensional Time Series," *IEEE Transactions on Visualization and Computer Graphics* 19, no. 3 (March 2013): 470–483. https://doi.org/10.1109/TVCG.2012.128
14. W. Derguech, E. Bruke, and E. Curry, "An Autonomic Approach to Real-Time Predictive Analytics Using Open Data and Internet of Things," in *2014 IEEE 11th International Conference on Ubiquitous Intelligence and Computing and 2014 IEEE 11th International Conference on Autonomic and Trusted Computing and 2014 IEEE 14th International Conference on Scalable Computing and Communications and Its Associated Workshops* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2014), 204–211.
15. A. Saxena and K. Goebel, *Turbofan Engine Degradation Simulation Data Set* (Moffett Field, CA: NASA Ames Research Center, 2008).
16. National Institute of Standards and Technology, "Cybersecurity Framework," National Institute of Standards and Technology, 2019. http://web.archive.org/web/20191030004314/https://www.nist.gov/cyberframework
17. N. Andrienko, T. Lammarsch, G. Andrienko, G. Fuchs, D. Keim, S. Miksch, and A. Rind, "Viewing Visual Analytics as Model Building," *Computer Graphics Forum* 37, no. 6 (September 2018): 275–299. https://doi.org/10.1111/cgf.13324
18. D. Xin, L. Ma, J. Liu, S. Macke, S. Song, and A. Parameswaran, "Accelerating Human-in-the-Loop Machine Learning: Challenges and Opportunities," in *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning* (New York: Association for Computing Machinery, 2018).
19. M. A. Nielsen, *Neural Networks and Deep Learning* (San Francisco, CA: Determination Press, 2015).
20. D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski, "Characterizing Guidance in Visual Analytics," *IEEE Transactions on Visualization and Computer Graphics* 23, no. 1 (January 2016): 111–120. https://doi.org/10.1109/TVCG.2016.2598468
21. C. Collins, N. Andrienko, T. Schreck, J. Yang, J. Choo, U. Engelke, A. Jena, and T. Dwyer, "Guidance in the Human–Machine Analytics Process," *Visual Informatics* 2, no. 3 (September 2018): 166–180. https://doi.org/10.1016/j.visinf.2018.09.003
22. W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang, "Recovering Reasoning Processes from User Interactions," *IEEE Computer Graphics and Applications* 29, no. 3 (May–June 2009): 52–61. https://doi.org/10.1109/MCG.2009.49
23. E. T. Brown, J. Liu, C. E. Brodley, and R. Chang, "Dis-function: Learning Distance Functions Interactively," in *IEEE Symposium on Visual Analytics Science and Technology* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2012), 83–92.
24. D. A. Keim, "Information Visualization and Visual Data Mining," *IEEE Transactions on Visualization and Computer Graphics* 8, no. 1 (January–March 2002): 1–8. https://doi.org/10.1109/2945.981847

25. J. D. Kennedy and M. Towhidnejad, "Innovation and Certification in Aviation Software," in *Integrated Communications, Navigation and Surveillance Conference* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2017), 3D3-1–3D3-15.
26. R. Abeyratne, *Strategic Issues in Air Transport: Legal, Economic and Technical Aspects* (Berlin, Germany: Springer, 2012).
27. J. Rush, "Super Puma that Crashed into the North Sea Killing 16 Men Was Declared Fit for Service the Day before the Tragedy, an Inquiry Has Heard," Daily Mail, 2014. http://web.archive.org/web/20191030004934/https://www.dailymail.co.uk/news/article-2535958/Super-Puma-crashed-North-Sea-killing-16-men-declared-fit-service-day-tragedy-inquiry-heard.html
28. R. Cabello, "Three.js.," 2010. http://web.archive.org/web/20191030005542/https://github.com/mrdoob/three.js
29. M. Bostock, V. Ogievetsky, and J. Heer, "D^3 Data-Driven Documents," *IEEE Transactions on Visualization and Computer Graphics* 17, no. 12 (December 2011): 2301–2309. https://doi.org/10.1109/TVCG.2011.185
30. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, et al., "Tensorflow: A System for Large-Scale Machine Learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (Berkeley, CA: USENIX, 2016), 265–283.
31. T. Dang, "Visualizing Multidimensional Health Status of Data Centers," in *Programming and Performance Visualization Tools* (Cham, Switzerland: Springer, 2017), 273–283.
32. L. Wilkinson, "Visualizing Big Data Outliers through Distributed Aggregation," *IEEE Transactions on Visualization and Computer Graphics* 24, no. 1 (January 2018): 256–266. https://doi.org/10.1109/TVCG.2017.2744685
33. T. N. Dang and L. Wilkinson, "TimeExplorer: Similarity Search Time Series by Their Signatures," in *International Symposium on Visual Computing* (Berlin, Germany: Springer, 2013), 280–289.
34. L. Wilkinson, A. Anand, and R. Grossman, "Graph-Theoretic Scagnostics," in *IEEE Symposium on Information Visualization* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2005), 157–164.
35. T. N. Dang and L. Wilkinson, "Transforming Scagnostics to Reveal Hidden Features," *IEEE Transactions on Visualization and Computer Graphics* 20, no. 12 (December 2014): 1624–1632. https://doi.org/10.1109/TVCG.2014.2346572
36. T. N. Dang and L. Wilkinson, "ScagExplorer: Exploring Scatterplots by Their Scagnostics," in *2014 IEEE Pacific Visualization Symposium* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2014), 73–80.
37. D. Holten, "Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data," *IEEE Transactions on Visualization and Computer Graphics* 12, no. 5 (September–October 2006): 741–748. https://doi.org/10.1109/TVCG.2006.147
38. R. Amar, J. Eagan, and J. Stasko, "Low-Level Components of Analytic Activity in Information Visualization," in *IEEE Symposium on Information Visualization* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2005), 111–117.
39. B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," in *IEEE Symposium on Visual Languages* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 1996), 336–343.
40. R. Borgo, J. Kehrer, D. H. S. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, "Glyph-Based Visualization: Foundations, Design Guidelines, Techniques and Applications," in *Eurographics* (Geneva, Switzerland: The Eurographics Association, 2013), 39–63.
41. B. C. M. Cappers, P. N. Meessen, S. Etalle, and J. J. van Wijk, "Eventpad: Rapid Malware Analysis and Reverse Engineering Using Visual Analytics," in *IEEE Symposium on Visualization for Cyber Security (VizSec)* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2018), 1–8.
42. D. K. Frederick, J. A. DeCastro, and J. S. Litt, *User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), NASA/TM—2007-215026* (Washington, DC: National Aeronautics and Space Administration, 2007).
43. E. Ramasso and A. Saxena, "Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets," *International Journal of Prognostics and Health Management* 5 (2014): 1–15.
44. T. Wang, J. Yu, D. Siegel, and J. Lee, "A Similarity-Based Prognostics Approach for Remaining Useful Life Estimation of Engineered Systems," in *International Conference on Prognostics and Health Management* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2008), 1–6.
45. F. O. Heimes, "Recurrent Neural Networks for Remaining Useful Life Estimation," in *International Conference on Prognostics and Health Management* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2008), 1–6.
46. L. Peel, "Data Driven Prognostics Using a Kalman Filter Ensemble of Neural Network Models," in *International Conference on Prognostics and Health Management* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2008), 1–6.
47. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation* 9, no. 8 (1997): 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
48. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature* 521, no. 7553 (May 2015): 436–444. https://doi.org/10.1038/nature14539
49. S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long Short-Term Memory Network for Remaining Useful Life Estimation," in *IEEE International Conference on Prognostics and Health Management* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2017), 88–95.
50. L. Jayasinghe, T. Samarasinghe, C. Yuen, J. C. N. Low, and S. S. Ge, "Temporal Convolutional Memory Networks for Remaining Useful Life Estimation of Industrial Machinery," *arXiv preprint arXiv:1810.05644*, 2018.
51. Interactive Data Visualization Lab, "Explainable Neural Networks for Multivariate Time Series," 2019. http://web.archive.org/web/20191030054904/https://idatavisualizationlab.github.io/V/RUL_Viz/
52. I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv preprint arXiv:1412.6572*, 2014.