On the Effectiveness of Recurrent Neural Networks for Live Modeling of Cyber-Physical Systems

Srikanth Yoginath, Varisara Tansakul, Supriya Chinthavali, Curtis Taylor, Joshua Hambrick, Philip Irminger and Kalyan Perumalla
Oak Ridge National Laboratory,
Oak Ridge, TN USA
Email: yoginathsb, tansakulv, chinthavalis, taylorcr, hambrickjc, irmingerp, perumallaks @ornl.gov

***Abstract*—Attention to cyber security of cyber-physical systems (CPS) has led to the development of innovative cyber-resilient methodologies to ensure early detection and mitigation of cyber anomalies and threats. The concept of *Digital Twin* (DT) has recently emerged as one of the approaches to achieve the objective of resilience. In the approach using DT, a software-based live model of a target CPS is used to continuously monitor, surveil and verify the correctness of the target CPS operation. In this paper, we empirically study the effectiveness of Recurrent Neural Network (RNN)-based models as the basis of DT-based resilience. We uncover the important characteristics of an RNN-based solution with experimentation on a lab-scale Canal Lock CPS emulator with live validations and attack scenarios. For the first time, we demonstrate actual, real-time use of a RNN-based model as a DT for performing live analysis on an operational CPS. Based on the observed results, we highlight the importance of a DT model's training interval, prediction interval and CPS polling interval in the process of anomaly detection. We uncover the limitations in anomaly detection due to real-time synchronization needs of the RNN-based DT. We highlight this uncovered tug of war between synchronization and anomaly detection is inherent in any complex CPS that is monitored and synchronized by relying on the same sensor streams of ground truth for both synchronization as well as anomaly detection.**

***Keywords*—Digital Twin, Digital Twin Framework, Resilient CPS, RNN models in CPS**

I. Introduction

Cyber Physical Systems (CPS) play a crucial role in sustaining and fulfilling our everyday needs as their application and utilization can be found in several critical infrastructures such as, energy generation/transmission/distribution, transportation, water treatment plants, nuclear reactors and so on. These systems are so critical that any degradation in their operations may have serious implications on the safety, security and economic stability of our society. The Department of Homeland Security (DHS) rightly identifies sixteen such CPS-based critical infrastructures [1]. The critical nature of large-scale CPS gives rise to variety of problems such as, device failures due to natural disasters, drifting sensors, misconfigured devices, and cyber security threats. One of the methodologies conceived to address the need of constant monitoring, surveillance and verification of operational correctness is the concept of Digital Twin (DT).

In one of the earliest works, Grieves et al. [2] define DT as a digital informational construct about a physical system that could be created as an entity of its own. This digital information would be a "twin" of the information that was embedded within the physical system itself and be linked with the physical system through the entire life cycle of the system. While Grieves et al. [2] discuss DT applicability over the entire life-cycle of a system, our work focuses on its applicability in live operation of CPS.

We use the concept of DT for monitoring, surveillance and verification of the target CPS. The idea is that the DT model, a perfect software-based replica of the real-life CPS, would execute in perfect time synchrony with its physical counterpart. This enables the living model that emulates the verifiably correct operational behavior of target CPS to detect any intentional or unintentional deviations of the target CPS operational behavior. Such dynamic monitoring and determination of deviation is helpful to detect anomalous behavior at its earliest occurrence.

Realizing such an accurate traditional simulation model that emulates the behavior of a continuously evolving large-scale industrial CPS system is challenging. This is because every CPS has its own unique characteristics based on the components used, logic applied and functionality realized. Thus, development of a simulation model would require manual tuning of the model parameters of every component model to replicate the exact physical behavior. Further, changes in the CPS due to hardware replacement or logic updates could alter the component behaviors requiring manual tuning of model parameters to reduce false alarms.

Deep-learning algorithms are shown to accurately learn non-linear function behaviors after significant training on CPS data. For example, a successful application of artificial neural networks [3] and recurrent neural network (RNN)-based [4] networks for anomaly detection in water purification CPS, under a wide-range of cyber attacks, can be found in the literature. However, they used the neural network models on collected data and *not* on the live system and hence, lack the real-time dynamics of using RNN as DT during CPS

operation.

In this paper, we develop an RNN model for an actual CPS test-bed, validate its performance in real-time during the CPS operation and evaluate the RNN model for such tasks. The real time application of the RNN model is the novelty of our work. We are not aware of any prior work that has applied RNN models to verify the operational behavior of a live CPS.

a) Organization: We discuss the development of our lab-scale canal lock CPS system emulator in Section II. In Section III, we discuss the development of the RNN model for the canal lock CPS. In Section IV, we validate the live operation and evaluate the RNN-based DT with respect to the canal lock CPS emulator in real-time and uncover the salient observations of our empirical study. We summarize the work and conclude in Section V.

b) Background and Related work: The concept of DT primarily originated within the manufacturing sector's Product Lifecycle Management (PLM) in 2002 as *Information Mirroring*, which tied the real and virtual spaces together [2]. As the concept of digital twins originated with the manufacturing sector, much of the literature has focused on the manufacturing applications [2], [5]–[8]. The US National Aeronautics and Space Administration (NASA) later adopted the PLM approach for its space systems development process [9]. Several other works on DT concept and design can be found at [10], [11] and [12]. Recently, software frameworks and relevant constructs for realizing DT have been put forth by others [13]. In power systems, the online simulators (such as the online version of DSA tools and PSS/E) that interact with SCADA systems in real-time [14] are closer to our work. However, while those rely on domain-specific simulations with a great amount of customized engineering knowledge, we use general-purpose RNN models as DT and our solution approach is hence significantly more generic.

II. CPS Emulation

To be able to experiment with a CPS in laboratory conditions and evaluate the utilization of RNN models as a DT, we developed a CPS emulation of Canal Lock CPS. In practice, a Canal Lock CPS is used to raise and lower boats or ships between stretches of water at different levels. The ship passes through a series of watertight chambers [15]. The entry to and exit from these chambers are controlled by gates as illustrated in Figure 1. In this simplistic illustration, a ship moves from a lower level to a higher level of water body by first entering to chamber1 through gate1. Chamber1's water level is increased to some intermediary level for the ship to cross over to chamber2 through gate2. Once the ship is in chamber2, the water level in this chamber is increased from the intermediary level to the higher water body level. At this point, the ship exits the chamber3 and the Canal Lock CPS through gate3. A similar sequence of steps in the reverse-order is followed when the ship needs to travel from the higher-level to the lower level of water body.

A. Canal Lock CPS Emulation Setup

The emulation of Canal Lock CPS was realized by using four acrylic tanks, labeled T1, T2, T3, and T4, as illustrated in Figure 1. The first tank T1 and the last tank T4 serve as lower-level and upper-level water bodies of the physical Canal Lock CPS, respectively. Tanks T2 and T3 act as the chamber1 and chamber2 of the Canal Lock CPS. Tanks T2 and T3 are equipped with pumps P2 and P3, and valves V2 and V3, respectively. The pumps are immersed in a reservoir, which holds water to fill all the tanks. Valves V2 and V3 drain water directly into the reservoir.

B. Hardware and Software

Each acrylic tank has the dimensions of $6 \times 6 \times 12$ $(length \times width \times height)$ $inches^3$. Each tank is fitted with an eTape™ liquid level sensor [16], which is designed to convert a resistance output of the liquid level sensor to a linear output voltage between 0 to 5V DC. The Schneider Electric Erie VM series Poptop™ series modulating valves [17] are used for draining water from tanks T2 and T3, which takes control the voltage of 0-10V DC. Two Diablo DC 3500 pumps [18] are used to pump water from the reservoir into tanks T2 and T3, and 5V DC supply is used to activate the pumps. The Allen-Bradley Micrologix 1100 PLC with two extension slots are used to maintain, control and emulate the Canal Lock CPS operational behavior. The liquid level sensors are connected to the analog input ports and the pumps are connected to the output ports of the PLC. We used the RSLogix 1100 Micro Starter Lite software to develop the ladder logic controller for the CPS.

C. Ladder Logic to emulate Canal Lock CPS

In Figure 2, we show the main loop of the ladder logic. The CPS can be enabled remotely over a Modbus connection from the Human-Machine Interface (HMI) or manually using the RSLogix software as shown in the first rung from the top.

The second rung checks for the emergency conditions, in presence of which the entire ladder logic would be deactivated. In the third rung, we update the status of all the peripherals namely, the sensors, pumps and valves. This subroutine only updates the statuses by writing into predefined data fields. For example, if pump P2 needs to be activated, its corresponding data field will be latched.

In the fourth rung, we process the events. For example, if the execution of third rung has set pump P2 status variable, then this would initiate the logic to signal a relevant PLC output port to actually turn on the pump P2. Similar conditions are used to control valves V2 and V3. However, valve control must be carefully regulated to control the undershoot or overshoot during draining. To achieve this, we applied Proportional Integral Derivative (PID) logic in the PLC. In the last rung before ending the main ladder, the ladder logic checks and executes a sequence of actions corresponding to up or down scenarios.

The ladder logic of up and down scenarios update predefined data fields, which are read after updating status in the third rung and acting upon the servicing of the requested events in the fourth rung. The upward and downward movements of the ship through the Canal Lock CPS emulator were realized by controlling the water levels of tanks T2 and T3,

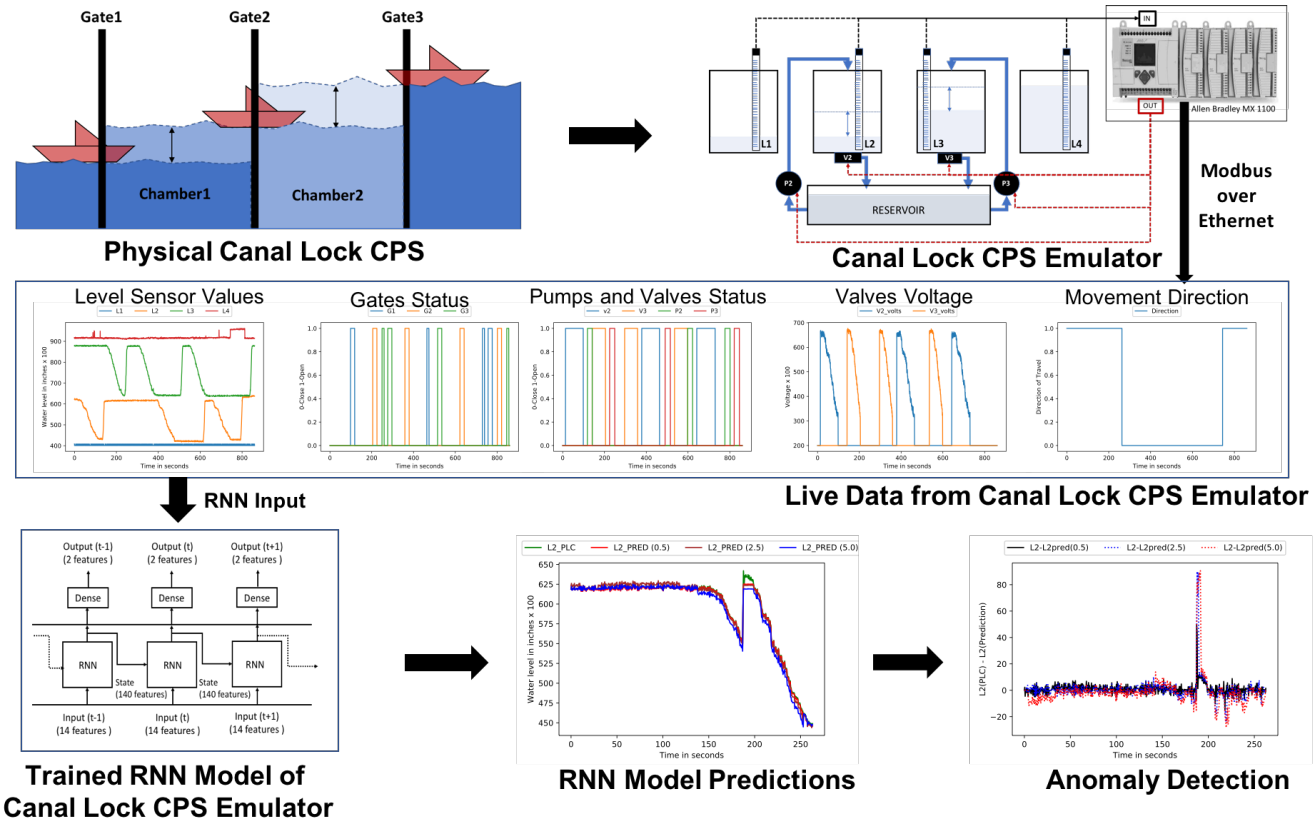


Fig. 1: Overview of the Canal Lock Case Study for RNN-based Live Modeling of CPS

corresponding to chamber1 and chamber2 of the physical Canal Lock CPS. The corresponding opening and closing operations of the gates, and the movement of ship from one tank to another is accommodated by using a constant ten second countdown timer. Based on where the ship arrives, the direction of its travel is determined. Hence, the arrival of ship at tank T1 suggests an up scenario or upward travel. Similarly, the ship's arrival at tank T4 suggests a down scenario or downward travel. The ladder logic is used to perform sequence of operations to emulate the Canal Lock CPS behavior for upward and downward movements of the ships through the canal and shown in Algorithm 1 and Algorithm 2, respectively. The ladder logic implementation of these algorithms is used by the PLC to control the water levels in the physical system emulating the behavior of the physical Canal Lock CPS.

D. Verification of Canal Lock Cyber-Physical System

The Canal Lock CPS can perform up and down movements of ships (two functions) and further, each function follows a different set of actions based on the initial state of the Canal Lock CPS. In this paper, we recognize these scenarios with combination of a *initial state* and *current operation* tuple. While the latter, *current operation*, is Up or Down movements, the former variable *initial state* represents the previous operation that was performed, whose end state is the initial state of the current operation (Up or Down). Hence, when the system operates continuously from a known physical state, there are only four possible combinations of scenarios

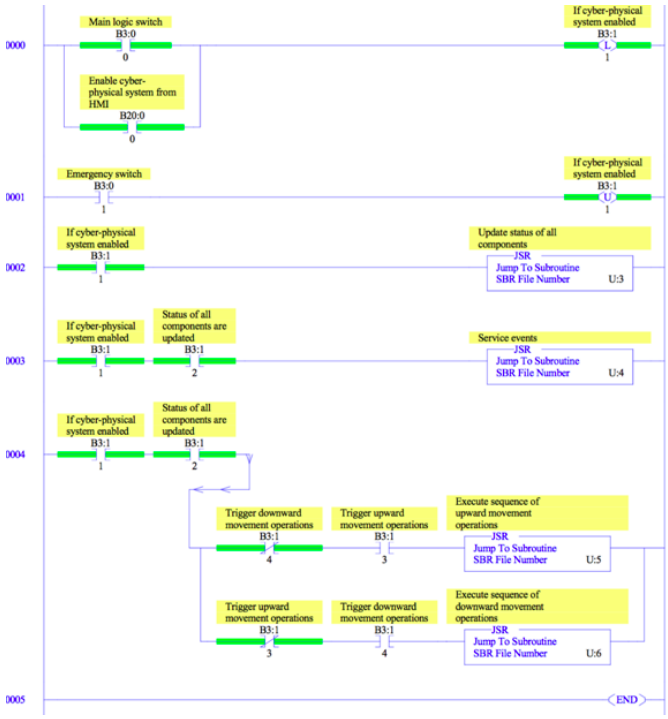


Fig. 2: Main Loop of Ladder Logic Implementation

with two different initial states for each function provided by the Canal Lock CPS.

Algorithm 1: Up Scenario

Data:

$L1$	Water level at T1 (constant)
$L4$	Water level at T4 (constant)
$L2$	Water level at T2
$L3$	Water level at T3
$L3_{low}$	Lowest expected water level at T3
$G1$	Gate between Tanks T1 and T2
$G2$	Gate between Tanks T2 and T3
$G3$	Gate between Tanks T3 and T4
$P2$	Pump in T2
$V2$	Valve in T2
$P3$	Pump in T3
$V3$	Valve in T3

Ship arrives at T1
Ensure all the gates $G1$, $G2$, and $G3$ are closed
Ensure $L2$ is at $L1$ using $P2$ and $V2$
Open gate $G1$
Ship moves to T2
Close gate $G1$
Increase $L2$ to $L3_{low} = (L4 - L1)/2$ using $P2$
Ensure $L3$ is at $L3_{low}$ using $P3$ and $V3$
Open gate $G2$
Ship moves to T3
Close gate $G2$
Increase $L3$ to $L4$
Open gate $G3$
Ship moves to T4
Close gate $G3$

Algorithm 2: Down Scenario

Ship arrives at T4
Ensure all the gates $G1$, $G2$, and $G3$ are closed
Ensure $L3$ is at $L4$ using $P3$ and $V2$
Open gate $G3$
Ship moves to T3
Close gate $G3$
Decrease $L3$ to $L3_{low} = (L4 - L1)/2$ using $V3$
Ensure $L2$ is at $L3_{low}$ using $P2$ and $V2$
Open gate $G2$
Ship moves to T2
Close gate $G2$
Decrease $L2$ to $L1$
Open gate $G1$
Ship moves to T1
Close gate $G1$

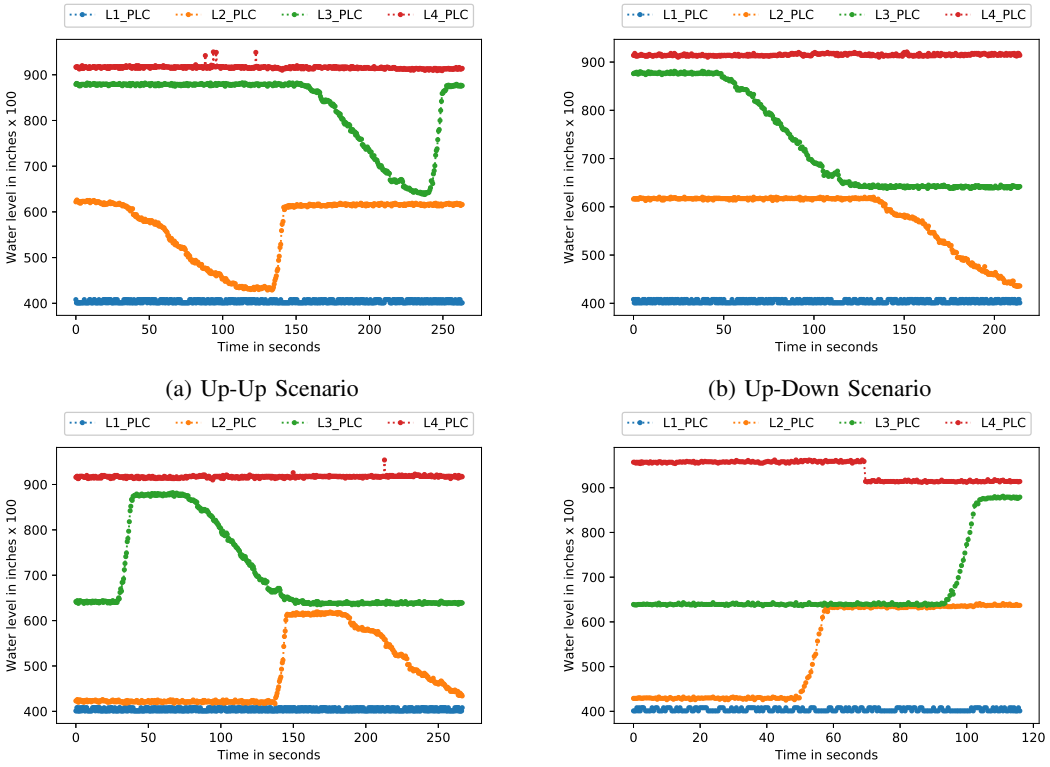
The combination of scenarios includes: (1) up followed by up (Up-Up), (2) up followed by down (Up-Down), (3) down followed by down (Down-Down), and (4) down followed by up (Down-Up). The water levels in tanks T1, T2, T3 and T4 are identified as L1, L2, L3 and L4, respectively. L1 and L4 represent the lower-level water body and upper-level water body of the canal system, respectively, and remain constant. The water levels are measured in inches. However, due to PLC limitations, water levels are transmitted between the PLC and DT as integers with 2 decimal precision (x100). We visualize this data using the raw numbers communicated from the PLC.

1) Up-Up Scenario: The plot in Figure 3a shows the variation of water levels in real-time for the Up-Up scenario, where the initial state of the water levels L2 and L3 are at $L3_{low}$ and L4, respectively, when the up scenario starts. As seen in Figure 3a, the water level L2 drops from its previous state ($L3_{low}$) to L1 water level which allows the ship to move across gate G1 in some constant time (ten seconds). After the ship moves to tank T2, the water level L2 is increased from L1 to $L3_{low}$ and the ship waits in tank T2 as the water level L3 decreases from its previous high position of L4. After the water level L3 drops to $L3_{low}$, then the ship moves from tank T2 to tank T3 across gate G2, as seen from Figure 3a at around 240 seconds. Then the water level L3 increases from $L3_{low}$ to L4 at which point the ship moves from tanks T3 to T4, thus exits the Canal Lock CPS.

2) Up-Down Scenario: The plot in Figure 3b shows the variation of water levels in real-time for the Up-Down scenario, where the initial state of water levels L2 and L3 are at $L3_{low}$ and L4, respectively, when the up scenario starts (same as Up-Up scenario). As seen in Figure 3b, the water level L3 is already at L4, so the ship easily transits from tanks T4 to T3 through gate G3. Once the ship is in tank T3, the water level starts to decrease from L4 to $L3_{low}$ from 50 to 120 seconds. When the water level L3 reaches $L3_{low}$, the ship transits from tanks T3 to T2 through gate G2 at around 130 seconds. Then, the water level L2 that is previously at $L3_{low}$ decreases to L1 at around 200 seconds, which is point the ship transits from tanks T2 to T1 through gate G1.

3) Down-Down Scenario: The plot in Figure 3c shows the variation of water levels in real-time for the Down-Down scenario, where the initial state of water levels L2 and L3 are at L1 and $L3_{low}$, respectively, when the down scenario starts. As seen in Figure 3c, the water level in tank T3 increases from $L3_{low}$ to L4 at around 50 seconds enabling the movement of ship to move from tanks T4 to T3 across gate G3. Then, the water level L3 slowly reduces from L4 to $L3_{low}$, which is also followed by increasing in the water level L2 from L1 to $L3_{low}$. At around 150 seconds, the ship transits from tanks T3 to T2 across gate G2, as seen in Figure 3c. Then, the water level L2 recedes from $L3_{low}$ to L1, thus allowing the ship to exit tank T2 and arrive at tank T1 at around 270 seconds.

4) Down-Up Scenario: The plot in Figure 3d shows the variation of water levels in real-time for the Down-Up scenario, where the initial state of water levels L2 and L3 are at L1 and $L3_{low}$, respectively, when the down scenario starts. This is by far the fastest running scenario because the initial state of water levels allow quick transition and the filling process for the tanks in our setup is faster than the draining process. As seen in Figure 3d, the ship transits from tanks T1 to T2 through gate G1 since the water levels of both tanks are at the water level L1. At around 60 seconds, the water level L2 starts increasing to reach $L3_{low}$ and at around 90 seconds, the ship transits from tanks T2 to T3 through gate G2. Following this, the water level L3 increases from $L3_{low}$ to L4 allowing the ship to reach its destination or tank T4 through gate G3



(a) Up-Up Scenario

(b) Up-Down Scenario

(c) Down-Down Scenario

(d) Down-Up Scenario

Fig. 3: Illustration of Canal Lock CPS emulation using level sensor values for all possible operation scenarios

at around 120 seconds.

III. RNN as the Digital Twin

In this section, we discuss the details of our RNN-based DT models, their training process, and live validation results.

A. Training Data

The data was collected at a rate of one set of readings per 0.5 seconds from the PLC. To emulate a Canal Lock CPS, we collected the data for the variables listed below. The plots corresponding to these fourteen variables is shown in Figure 1.

(a) Water levels L1, L2, L3, L4 in tanks T1, T2, T3 and T4, in inches (float values between 0-10 inches)
(b) Gates G1, G2, G3 between tanks T1 and T2, T2 and T3, and T3 and T4, respectively (0 - Closed, 1 - Opened)
(c) Status of pumps P2 and P3 of tanks T2 and T3 (0 - OFF and 1 - ON)
(d) Valves V2 and V3 of tanks T2 and T3 (0 - OFF and 1-ON)
(e) Control voltages Vv2 and Vv3 on valves V2 and V3, in Volts (float values between 0 – 10 V)
(f) Direction D indicates the up or down scenario (0 – DOWN and 1 – UP)

B. RNN Model

We trained the model using all fourteen collected features. We used 140 (14 × 10) units resulting in an RNN model

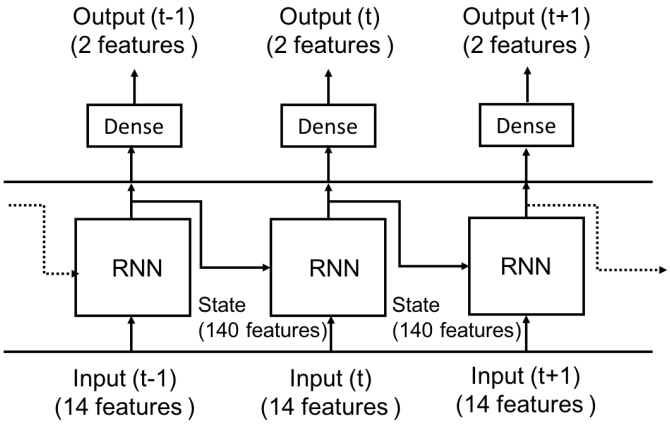


Fig. 4: Unrolled RNN Model

learning parameters of 21,700 ((14 features + 140 units) × 140 + 140 bias). This RNN layer is followed by a fully-connected dense layer with 140 inputs, and two outputs, which learns (280 weights + 2 bias) parameters. In total, the whole network learns 21,982 (21,700 + 282) variables. The two predicted outputs were the expected water levels *L2'* and *L3'*.

In Figure 4, we show the unrolled RNN model that works on a time sequence of inputs. In addition to the two feature outputs generated for every input, the RNN layer also generates the state data that is passed to the next sequence of RNN computation. In our model, we used 140 units, which

correspond to ten previous states, to compute the current state. This state information is passed through a dense layer to predict two output values namely, water levels of tanks T2 (L2) and T3 (L3). The input data collected was normalized to range between 0 and 1, using the minimum and maximum values of the collected features. The same minimum and maximum values were used to re-scale the predicted values back to Canal Lock CPS emulator relevant values.

A mean-square-error loss along with an Adam [19] optimizer was used to learn a total of 21,982 parameters. The models recorded are based on the best loss value observed across different epochs. The maximum number of epochs to run was set to 100. We applied the early stopping option in Keras library which terminated the training ahead of 100 epochs when repeatedly consistent loss values were observed. All the trained models had loss less than 0.1 percent.

C. Test Models

As mentioned previously, four different scenarios of executions namely, Up-Up, Up-Down, Down-Down and Down-Up, are possible in our Canal Lock CPS. To find the minimal set of data that are required to build an RNN model that would efficiently predict the Canal Lock CPS variables (water levels L2 and L3) during the up and down scenarios, we created two RNN models.While the first model (*RNN Model-1*) uses data from all the possible scenarios of Canal Lock CPS, the second model (*RNN Model-2*) uses data from Up-Down (Figure 3b) and Down-Up (Figure 3d) scenarios only. However, both these scenarios together capture the transition dynamics of water movements between the two water levels in both tanks T2 and T3. Since the transition dynamics between water levels remains the same across scenarios, the hypothesis of the *RNN Model-2* is that only the data from the Up-Down and Down-Up scenarios are sufficient to predict the water level transition dynamics across all possible scenarios of the Canal Lock CPS emulator.

RNN Model-1 and *RNN Model-2* are trained to predict the very next point ($L2'_{0.5}$, $L3'_{0.5}$), 5 polling points ahead ($L2'_{2.5}$, $L3'_{2.5}$) and 10 polling points ahead ($L2'_{5.0}$, $L3'_{5.0}$), which are approximately 0.5, 2.5 and 5.0 seconds in future, respectively.

IV. Digital Twin Evaluation

A. Live Validation

To validate the developed models, we executed a continuous sequence of functional scenarios (Up-Down-Down-Up-Up) back-to-back, starting with the end of an initial UP scenario whose data was not recorded. As this validation scenario ran on the PA, we collected the data from the PLC using Modbus over Ethernet, and this data is used for live prediction. In between every polling interval, the data read from PLC was input to six models (*RNN Model-1* and *RNN Model-2*) predicting L2 and L3 values for 0.5, 2.5 and 5.0 seconds ahead of time. The delay between individual polling was less than 0.5 seconds.

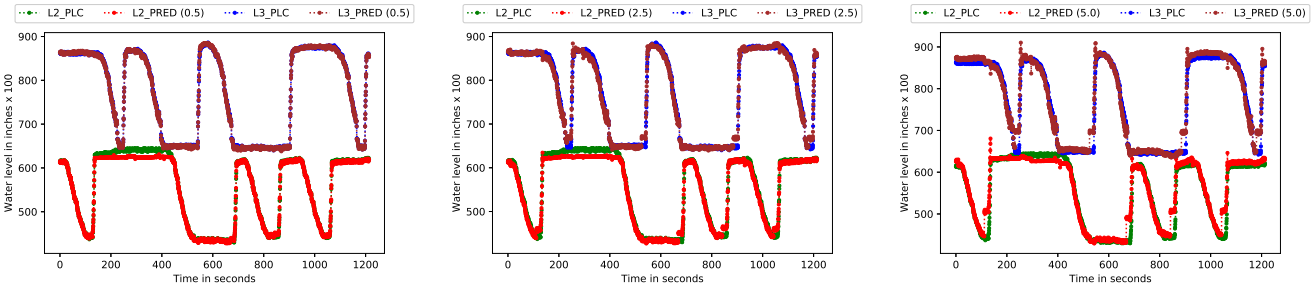
Figure 5a, Figure 5b and Figure 5c show *RNN Model-1*'s performance in predicting L2 and L3 values expected in 0.5, 2.5 and 5.0 seconds in the future. In each of these plots, the observed L2 and L3 values from the PLC are compared against the predicted values. The points within each plot hold a one-to-one correspondence of the predicted value to what actually was observed when real time catches up to the predicted time. There exists a close correspondence between the predicted and observed points as seen from the Figure 5a. However, this closeness of match slightly deteriorates when the prediction time increases from 0.5 to 5.0 seconds. These distortions are more pronounced during the sudden, rapid transitions in water level, which occurs due to the active pump filling the tanks.

Similarly, Figure 6a, Figure 6b and Figure 6c show RNN Model-2's performance in predicting L2 and L3 values expected in 0.5, 2.5 and 5.0 seconds in future. With the minimal set of data used for training in the RNN model-2, it is interesting to note that we are able to observe good results for next point predictions. However, they slightly deteriorate as the prediction time increases from 0.5 seconds to 5.0 seconds. The RNN Model-2 captures the dynamics of draining and filling of tanks very well. However, its predictions are relatively deteriorate in the flat regions; this is more pronounced when the prediction time is 2.5 and 5.0 seconds.

B. Anomaly Detection in Canal Lock CPS

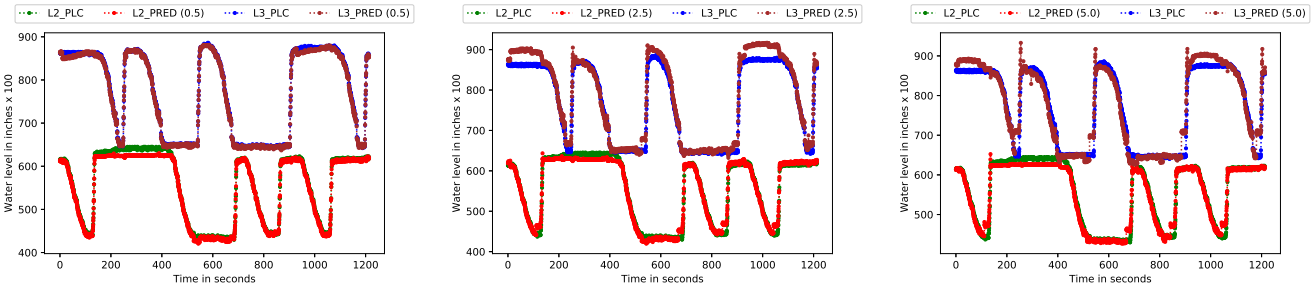
1) Algorithm: We employ a simplistic difference-based method to visually segregate outliers and hence, evaluate the ability to detect anomalies using DT. The anomaly detection pseudo-code is shown in Algorithm 3. The PLC value (γ) read is compared with the previously predicted value (ρ). The difference between γ and ρ captures deviations with respect to DT predictions in real-time. To compute the difference between the predicted and actual values whose prediction time is more than the polling interval (0.5 seconds), we initially wait for n polling intervals to reach the prediction time interval before calculating the differences. The difference is calculated by negating the previously predicted value from the new reading from the PLC. If the difference is positive, then the value read from the PLC is greater than the model predicted value and the reverse is true if the difference is negative. The idea here is that the observed values are expected to be closer to the expected values and hence, the difference should be close to zero. Anything contrary to this rule would qualify as an outlier indicating an attack or a false positive.

2) Attack Scenarios: To evaluate the anomaly detection capability using the DT, we tested it in the Up-Down and Down-Down scenarios. The normal functioning of the Up-Down scenario is shown in Figure 3b, where the water levels in both tank T2 and tank T3 drop from their initial levels to accomplish the movement of the ship from tanks T4 to T1. The Down-Down scenario also accomplishes the same movement of ship from tanks T4 to T1, but it starts with a different initial state. Due to this change in the initial state, water is pumped in both the tanks before draining, as shown in Figure 3c. An attack on the CPS was realized by introducing an unexpected inflow of water in tank T2 while it is in the process of being drained. This abnormal inflow creates an unexpected and abrupt increase in the water level in tank T2, and this deviation from the expected behavior result in significant outliers suggesting anomaly.



(a) 0.5 second ahead predictions (b) 2.5 seconds ahead predictions (c) 5.0 seconds ahead predictions

Fig. 5: Live validation performed using *RNN Model-1* on the operational Canal Lock CPS emulator results in predicting expected L2 and L3 values for different prediction intervals. The predictions closely correspond to the real time observations. The larger interval predictions are slightly erroneous during sudden increase in the water levels due to the pump activity.



(a) 0.5 second ahead predictions (b) 2.5 seconds ahead predictions (c) 5.0 seconds ahead predictions

Fig. 6: Live validation performed using *RNN Model-2* on the operational Canal Lock CPS emulator results in predicting expected L2 and L3 values for different prediction intervals. In addition to the erroneous predictions during pump activity as seen with *RNN Model-2*, they also deteriorate with larger prediction intervals especially in the regions where the water levels are relatively constant.

Algorithm 3: Anomaly detection algorithm

Data:
- γ Data read from PLC
- ρ Next point data predicted using γ
- δ Difference between γ and ρ
- ε End of validation
- n number of points ahead in future

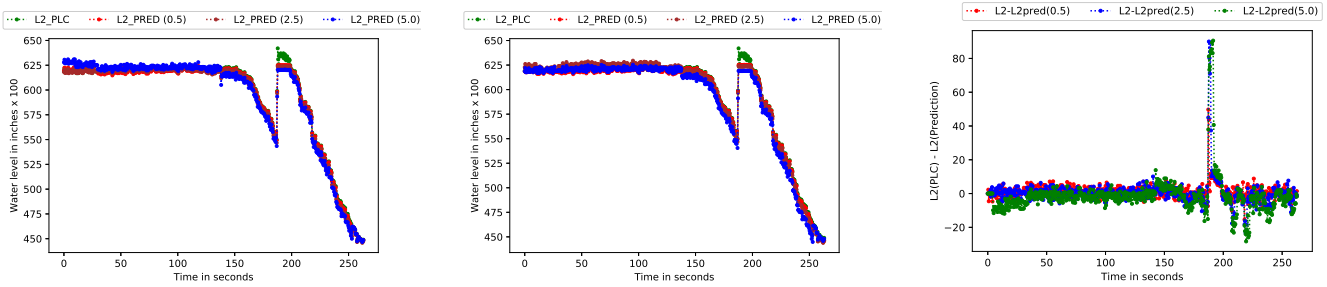
for i *in range(1:n)* **do**
  $\gamma_i \leftarrow read\ PLC$
  $\rho_i \leftarrow predict\ using\ \gamma_0$
end
for j *in range(n:*ε*)* **do**
  $\gamma_j \leftarrow read\ PLC$
  $\delta_j = \gamma_i - \rho_{(j-n)}$
  $\rho_j \leftarrow predict\ using\ \gamma_i$
end

3) Anomaly Detection in Up-Down Attack Scenario: The effects of the attack in the L2 values in the Up-Down scenario and their corresponding predictions using *RNN Model-1* and *RNN Model-2* are shown in Figure 7a and Figure 7b, respectively. In these plots, the predicted values of $L2'_{0.5}$, $L2'_{2.5}$ and $L2'_{5.0}$ are placed alongside the L2 value used to predict them. Thus, $L2'_{5.0}$ appears to lead $L2'_{2.5}$, which leads $L2'_{0.5}$ to almost overlap with the L2 value read from the PLC.

A spike representing an abrupt increase in L2 while tank T2 is draining is seen in both Figure 7a and Figure 7b. These are the due to the presence of outliers in the difference dataset. Further, the predictions from different models appear almost similar to each other. This can be expected as the attack is happening during transition, where both models deliver good predictions. In Figure 7c, we plot the difference between L2 and its corresponding predictions (Algorithm 3), showing an unexpected sharp spike at the time of attack. However, such spikes are very narrow because the RNN models predict based on recently received ground truth values. As the RNN model predictions are based on the recently read values, hence the predictions quickly catch up with the L2 trends after the attack. They differ only during the time of attack or during sudden abrupt state changes in the system.

4) Anomaly Detection in Down-Down Attack Scenario: This attack on the tank T2 in the Down-Down scenario is similar to the of Up-Down scenario. However in the Down-Down scenario, the fast filling of tank T2 precedes its draining. In the Figure 8a and Figure 8b, we show the L2 (PLC and prediction) trends and the corresponding differences based on Algorithm 3, respectively, for the Down-Down attack scenario.

As seen in Figure 8b, the legitimate behavior tank T2 water level raising results in the significant outliers in predictions with larger prediction interval (2.5 and 5.0 second) resulting

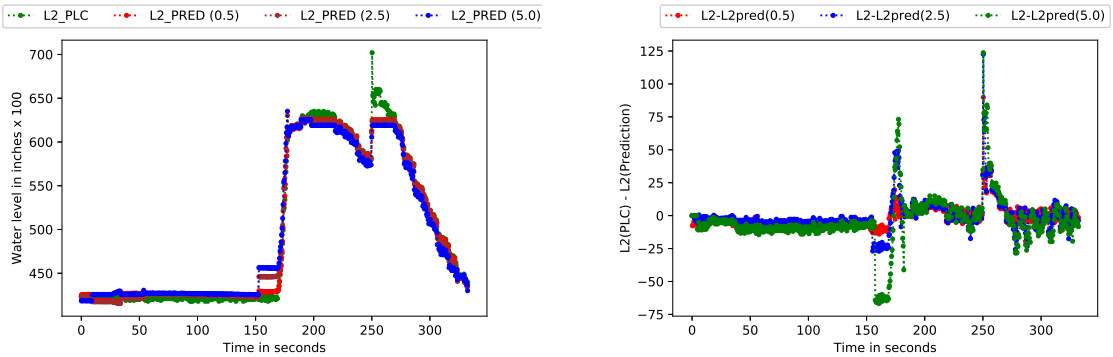


(a) The L2 values are compared with *RNN Model-1* L2 predictions

(b) The L2 values compared with *RNN Model-2* L2 predictions

(c) Anomaly estimation using live PLC values and the *RNN Model-2* predictions

Fig. 7: Attack on Tank T2 in the Up-Down scenario of Canal Lock CPS emulation, results in abrupt changes in L2 sensor values. The L2 values are compared with the 0.5, 2.5, 5.0 seconds *RNN Model* predictions. The difference algorithm 3 is used to determine anomalies.



(a) The L2 values compared with *RNN Model-2* L2 predictions

(b) Anomaly estimation using live PLC values and the *RNN Model-2* predictions

Fig. 8: Attack on Tank T2 in the Down-Down scenario of Canal Lock CPS emulation, results in abrupt changes in L2 sensor values. The L2 values are compared with the 0.5, 2.5, 5.0 seconds *RNN Model-2* predictions. The difference algorithm 3 is used to determine anomalies.

in false-positives.

C. Observations

a) Several Time-Intervals: Three different time-intervals can be observed in the live DT operation. They are (a) *Training-Interval* the time-interval between consecutive points of the real time data used for training (b) *Prediction-Interval*, is a point in time in future at which the model predicts values, it could be next point or some point in distant future and (c)*Polling-Interval* the time interval between two consecutively polled data points from the PLC of the target CPS during live operation.

The *Training-Interval* limits the precision with which the model can detect anomalous behavior. Hence, the time-series data used for RNN model training must ensure that this time-interval should be lesser than the distortion representative with lowest time-interval. Hence, DT's ability can be limited to or expanded to many sensor variables by changing this time-interval. Further, in our experiments and training we have considered a constant value of 0.5 seconds for the *Training-Interval*. However in practice the data collected for training by polling the PLC is based on the turn around time to obtain values from the PLC, which we observed not to be constant value but a period itself. We also observed that such variance of this period becomes larger when polling is performed back to back without any collector induced delay. The *Prediction-Interval* can provide next-point of predictions or distant-future (several points) predictions. The distant-future predictions are more sensitive to sudden input changes and hence are good for anomaly detection. However, their sensitivity is also a source for false-positives. The *Polling-Interval* should at least be equal to *Training-Interval*. A lower time-interval than *Training-Interval* should be able to better detect an anomalous activity. However, a lower Polling-Interval will also increase the DT load on PLC.

b) Synchronization effects on Anomaly Detection: The DT's capability to predict accurately is limited by the *Polling-Interval* at which point a newly read point from the PLC resets the model. We refer to this as the synchronization point, where the DT time-lines are synchronized with the target CPS time-line. Unlike simulations the RNN model predictions are based-off the value it just read and hence, its capability to detect the anomaly is limited to this *Polling-Interval*. Beyond this period, if the next read point falls within the awareness of the learned system than predictions

are sensible. However, if the input does not fall under the learned system state-space the predictions could be random. In the attack cases that we discussed, the post-attack state falls within the learned RNN-model and hence the post-attack predictions are reflective of system behavior after the attack, as seen in Figure 7c and Figure 8b. Hence, this leads to a situation where if the synchronization-period is very small, the anomalous events might just appear as jitter and if its too large than we can completely miss the anomalous event altogether. The predictions with varying *Prediction-Intervals* could help in such cases. Alternatively, the use of simulations that are not tightly bound by the synchronization needs might be able to detect the anomalous behavior better.

c) Emergent Behaviors with Simulations: A well trained RNN models would serve as a good DT with limited capabilities. While RNN model might be able to detect deviations from normal behavior, it will not be able able to predict the future emergent behaviors of the CPS behavior. Unlike the RNN models, the emergent behaviors can predicted using simulations. We are currently investigating the utilization of RNN-models with simulation models as a part of our future work.

V. Summary and Conclusion

In the research presented in this paper, we emulated the Canal Lock CPS behavior physically within our laboratory by using an actual PLC that is used in an industrial setup along with several sensors, pumps and valves. Within our controlled CPS setup, we identified all the possible Canal Lock CPS operational scenarios. From the Canal Lock CPS emulation data, we trained a set of RNN Models to use them as the core of a DT, and performed live validation of the system to evaluate our RNN models. We devised attacks on the Canal Lock CPS emulation system and evaluated DT concept for anomaly detection capability using a simple difference-based algorithm. The work performed in this paper has been summarized in Figure 1. In our salient observations, we highlighted the importance of tuning prediction intervals used in the RNN models and the polling intervals used to obtain real-time values from the PLC, for better anomaly detection and to reduce false positives. We also highlighted that even though RNN model predictions follow the PLC time-series values, they would not be able to capture the emergent behaviors of the complex system, for which simulations are needed. In future, we intend to evaluate the possibility of using module-specific RNN models, combining RNN models with discrete event simulation models, and extend our work to evaluate the RNN-based models on a large scale real-life CPS.

References

[1] "Department of homeland security critical infrastructure sectors," https://www.dhs.gov/critical-infrastructure-sectors, 2018.

[2] M. Grieves, "Product lifecycle management," www.egr.msu.edu/classes/ece480/goodman/PLMinEngineering.ppt, 2002.

[3] A. A. Abokifa, K. Haddad, C. Lo, and P. Biswas, "Real-time identification of cyber-physical attacks on water distribution systems via machine learning-based anomaly detection techniques," *Journal of Water Resources Planning and Management*, vol. 145, no. 1, p. 04018089, 2019.

[4] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *High Assurance Systems Engineering (HASE), 2017 IEEE 18th International Symposium on*. IEEE, 2017, pp. 140–145.

[5] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," 2014, white paper.

[6] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Springer, Cham, 08 2016.

[7] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in cps-based production systems," *Procedia Manufacturing*, vol. 11, pp. 939–948, 2017.

[8] *Micro Manufacturing Unit and the Corresponding 3D-Model for the Digital Twin*, Procedia Manufacturing. Stockholm, Sweden: Elsevier, May 2018.

[9] P. Caruso, D. Dumbacher, and M. Grieves, "Product lifecycle management and the quest for sustainable space exploration," in *AIAA SPACE 2010 Conference & Exposition*, 2010.

[10] B. Ferguson, A. Tall, and D. Olsen, "National cyber range overview," in *Military Communications Conference (MILCOM), 2014 IEEE*. IEEE, 2014, pp. 123–128.

[11] S. Glaessgen, "The digital twin paradigm for future nasa and u.s. air force vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. American Institute of Aeronautics and Astronautics, April 2012.

[12] E. Tuegel, A. Ingraffea, T. Eason, and M. Spottswood, "Reengineering aircraft structural life prediction using a digital twin," *International Journal of Aerospace Engineering*, vol. 2011, p. 154798, 2011.

[13] M. Eckhart and A. Ekelhart, "Towards security-aware virtual environments for digital twins," in *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*. ACM, 2018, pp. 61–72.

[14] C. Dufour, Z. Soghomonian, and W. Li, "Hardware-in-the-loop testing of modern on-board power systems using digital twins," in *2018 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*. IEEE, 2018, pp. 118–123.

[15] Wikipedia, "Lock (Water Navigation)," https://en.wikipedia.org/wiki/Lock_$water_navigation$, 2018.

[16] I. Milone Technologies, "etape continuous fluid level sensor pn-12110215tc-x," http://www.original.milonetech.com/uploads/Standard_eTape_Datasheet.pdf, 2018.

[17] S. Electric, "Erie VM Series Poptop^{TM} Series Modulating Valves," https://iportal2.schneider-electric.com/Contents/docs/F-26801-8.PDF, 2018.

[18] "Reef octopus diablo variable speed water pump dc-3500," https://www.marinedepot.com, 2018.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980