Development of Cyber Physical Layer to Defend EV Systems from Cyber Attacks: A Digital Twin Approach

Deepi Singh
Department of
Electrical and Computer Engineering
SUNY Stony Brook
Stony Brook, NY
Email: deepi.singh@stonybrook.edu

Reetam Mandal
Department of
Electrical and Computer Engineering
SUNY Stony Brook
Stony Brook, NY
Email: reetam.mandal@stonybrook.edu

Yash Patel
Department of
Electrical and Computer Engineering
SUNY Stony Brook
Stony Brook, NY
Email: yash.patel.1@stonybrook.edu

***Abstract*—The ubiquitous influence of power electronics in electric vehicles (EV) translates to increased vulnerabilities and chances of cyber-attacks. Moreover, the Industry 4.0 is transforming motor drives into intelligent and rapidly evolving edge devices with formidable computational power, thus enabling advanced sensing and Seamless external sensor connectivity. Nonetheless, this cyberspace provides numerous opportunities for malicious attacks threatening the security of the EVs and their applications, potentially resulting in accidents, injuries, property/infrastructure damages, even taking human lives. In this project, we analyze emerging power electronics security challenges on the controller of the power electronics drive and proposed a digital twin-based approach for the secure and dependable operation of the system. The approach considers developing a digital twin-based intrusion detection mechanism to be deployed at the electric motor drive level such that it can deal with malicious data/control commands initiated by attacks on the controller. The HIL (Hardware-in-Loop) based simulation is conducted to quantitatively analyze the impact under multiple attack scenarios on the controller for electric drive systems in electric vehicles and then they are classified using bagging ensemble based Machine Learning.**

***Index Terms*—DC-DC Buck Converter, Electric Motor Drive, Digital Twin, Cyber Security, Cyber Physical System, Hardware Security, AI Classficaton**

I. Introduction

With the growing penetration of Internet-of-Things enabled applications like electric vehicles (EVs), power electronics systems are becoming more vulnerable to cyber-physical attacks. EVs have been transforming modern transportation and energy systems. The mass adoption of EVs has been steadily increasing for the past two decades across all domains. Considering fuel savings and environmental impacts, EVs are perfectly aligned. Hence, automotive systems are going through massive transformation to increase vehicle safety, efficiency, and reliability to realize EVs . Meanwhile, due to the lack of cyber awareness in the power electronics community, it becomes more urgent to develop monitoring and diagnosis strategies for networked power electronics systems. For many safety-critical applications, if these threats are not detected in the early stage, they can lead to a catastrophic failure and substantial economic loss.Recently, some preliminary works on electric vehicle charging cyber security have been published, such as [1]–[5]. However, none of them investigates the impacts of cyber attacks on power electronics and electric drive systems (EDSs). In [11], a CPS model of EDSs in an EV is introduced, based on which, the impacts of various data integrity attacks on the predefined performance metrics are analyzed. Due to the increased cyber threats on physical systems [6], cyber physical systems (CPS) models have been developed to investigate the interaction between physical systems and cyber systems in smart grids [7]–[9], which can then be used to assess the operational reliability and vulnerability, the transient angle and voltage stability, and the frequency and electricity market operation due to cyber attacks [10].

While the current state of CPS security is already strained, smart technology trends proceed to evolve, pushing traditional protection mechanisms to their limits. As a result, new methods to support the implementation of a holistic security approach are needed. Considering the interdependency of the cyber and physical domains in which these systems function, adequately protecting CPSs represents a pressing challenge. Digital twin is a rather new concept in industry. With digital twins, we have virtual replicas of physical systems so that they precisely mirror the internal behavior of the physical systems [12].[13] discusses how a digital twin replication model and corresponding security architecture can be used to allow data sharing and control of security-critical processes. In [14] The Digital twin based intrusion detection algorithm is proposed for industrial control system.In this paper, we briefly discuss the emerging power electronics security challenges in EVs and the potential countermeasure approaches and their shortcomings. We simulated multiple attacks on the controller of Electric Drives for our case study. Then we proposes a digital twin based intrusion detection algorithm to be developed at the power electronic level. Finally, a Machine Learning based classifier is used to classify all these attacks.The main contributions of this article are listed as follows:

- Cyber security for hardware and communication modules using real time computation using a digital twin approach rather that the contemporary fault checking using a simulation based approach.
- Identification of threats including motor speed and other backdoor channels in sensors for injection of malware, coolant/temperature variation by valve malfunctioning and battery thermal stability threats, battery potential infringement, show speed change, tyre pressure based motor power variation/ brake fluid pressure based failure of braking, controller parameter variations and compromise of charging station that could cause catastrophic failure of power grid operations.
- Cloud based heuristics for classification and notification to the users.

II. System Theory and Modelling

A. Digital Twin

The idea of a digital twin comes into picture due to the number of permutations and combinations of various inputs and the system of systems in an electric vehicle. The math, computation and time behind computing every possible scenario is extremely vast and the bottleneck in such a scenario is data speeds and computation via quantum computing to keep it real time. Hence in order to simplify and make this a little in scale the digital twin approach holds good and provides enough results in a decent time to mitigate/ reduce errors. The key components that play a key role in the principle of digital twinning are as follows:

- Physical System: A real physical object or a process.
- Sensors: The components used to capture operational signals of a physical system like pressure, voltage, temperature, current etc.
- Data: The data from the sensors represent the operating conditions or characteristics of a physical systems and they are used to update the digital twin continuously.
- Digital Twin: The digital replica of a physical system.
- Analytics: Analytic techniques are used to process the data from sensors by using advanced algorithms and make the insights of digital twin same with the insights of its physical counterpart as close as possible.
- Insight: The inside information of a digital twin. The insights could be used to guide the action of physical system and improve its performance. Also we use it to assess systemic operations of the physical system.
- Action: According to the insights from analytics, action could be performed to better the performance of physical system or protect the physical system.

1) Digital Twin of a Buck Converter and Buck Converter Modelling:

The Buck Converter is used in SMPS circuits where the DC output voltage needs to be lower than the DC input voltage. The DC input can be derived from rectified AC or from any DC supply. It is useful where electrical isolation

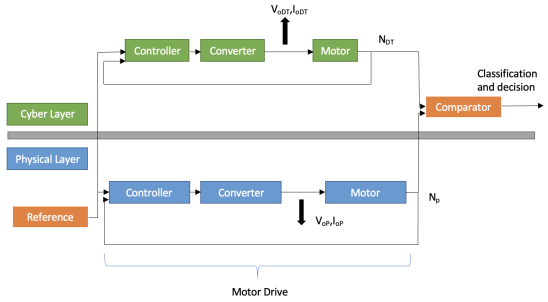

Fig. 1. Abstract Model

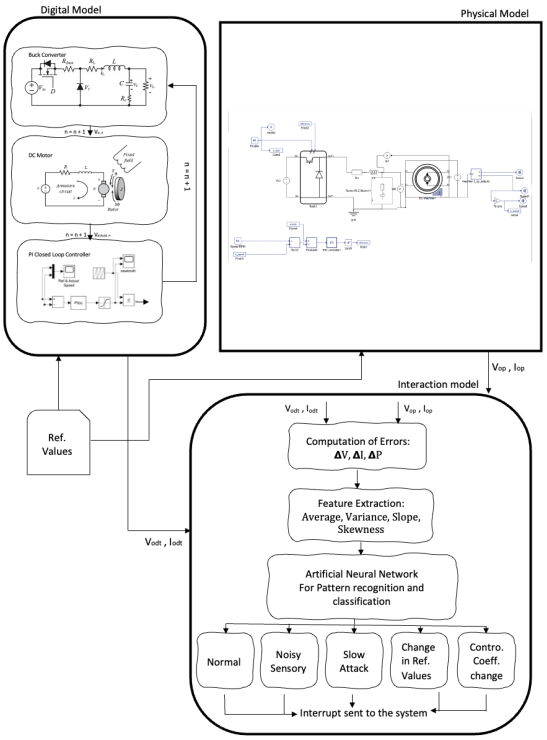

Fig. 2. Conceptual Flowchart of Implementation

is not needed between the switching circuit and the output, but where the input is from a rectified AC source, isolation between the AC source and the rectifier could be provided by mains isolating transformer.

The figure shows the equivalent circuits of buck converter operating at on and off state and represented by the state space equations where where i_L is the inductor current, Vo is the output voltage and v_C is the capacitor voltage; R_{dson}, R_L and R_C are the parasitic resistances of MOSFET, inductor and capacitor, respectively; v_{in} is the input voltage and V_f is the forward voltage of diode; D is 1 when the MOSFET is on and 0 when the MOSFET is off.

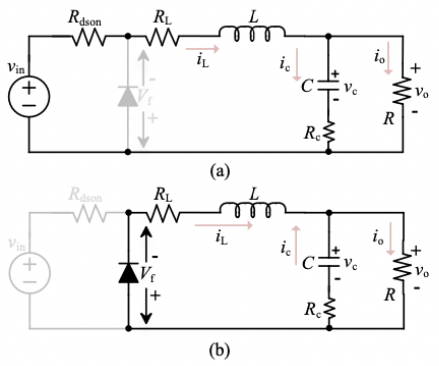

Fig. 3. Equivalent Circuit of a Buck Converter: (a) MOSFET on state (b) MOSFET off state

$$\begin{bmatrix} \frac{di_L}{dt} \\ \frac{dv_C}{dt} \\ v_o \end{bmatrix} = \begin{bmatrix} \frac{-1}{L}A & \frac{-1}{L}\frac{R}{R_C+R} \\ \frac{1}{C}\frac{R}{R_c+R} & \frac{-1}{C}\frac{1}{R_c+R} \\ \frac{R_cR}{R_c+R} & \frac{R}{R_c+R} \end{bmatrix} \times \begin{bmatrix} i_L \\ v_C \end{bmatrix} + D \begin{bmatrix} \frac{1}{L}V_{in} \\ 0 \\ 0 \end{bmatrix}$$

$$A = DR_{dson} + R_L + \frac{R_CR}{R_C+R}$$

Two ways can be applied to solve equation for A, and obtain i_L and v_C. One is to calculate the eigenvector and eigenvalue of differential equations and construct the general solution. Then, by using the initial values of i_L and v_C, the specific solution of these differential equations can be obtained. This method demands heavy computation, especially the calculation of eigenvector and eigenvalue. The other one is to linearize the differential equations with acceptable accuracy, which is used in this paper. Then, the output voltage v_o can be described with discrete time step:

$$v_{o,n+1} = i_{L,n+1}\frac{R_CR}{R_C+R} + v_{c,n+1}\frac{R}{R_C+R}$$

where the n^th time step is defined as the present time interval, the $(n+1)^{th}$ time step represents the next one. $v_{o,n+1}$ indicates the output voltage at $(n+1)^{th}$ step, which is unknown at present n^{th} step. Therefore, in the following discussions, $i_{L,n+1}$ and $v_{c,n+1}$ are derived based on the present values $i_{L,n}$ and $v_{c,n}$, so that the output voltage at the $(n+1)^{th}$ time step can be represented by the inductor current and capacitor voltage at n^{th} time step. Runge-Kutta is a typical method to solve differential equations. A typical 4^{th}-order Runge-Kutta method is used in this paper to linearize the differential equations and it is considered as sufficient for buck converter modeling to achieve an negligible error.

The above two are separate state space equations for ON/OFF conditions of the Pulse Width Modulation (PWM). Runge Kutta (RK) method is generally used to solve complex 4th order differential equations. In this case the equations of the current and output voltage are 4th order differential equations and need to be linearized and solved using RK method. The RK equations can be listed as follows:

$$i_{L,n+1} = i_{L,n} + \frac{h}{6}(k_{a1} + 2k_{a2} + 2k_{a3} + 2k_{a4})$$

$$v_{c,n+1} = v_{c,n} + \frac{h}{6}(k_{b1} + 2k_{b2} + 2k_{b3} + 2k_{b4})$$

where k_{a1} - k_{a4} and k_{b1} - k_{b4} are used to calculate the average change rate between $(n)^{th}$ and $(n+1)^{th}$ step as show below:

$$k_{a1} = f_1(x_n, y_n)$$
$$k_{b1} = f_2(x_n, y_n)$$
$$k_{a2} = f_1(x_n + \frac{h}{2}k_{a1}, y_n + \frac{h}{2}k_{b1})$$
$$k_{b2} = f_2(x_n + \frac{h}{2}k_{a1}, y_n + \frac{h}{2}k_{b1})$$
$$k_{a3} = f_3(x_n + \frac{h}{2}k_{a2}, y_n + \frac{h}{2}k_{b2})$$
$$k_{b3} = f_3(x_n + \frac{h}{2}k_{a2}, y_n + \frac{h}{2}k_{b2})$$
$$k_{a4} = f_4(x_n + hk_{a1}, y_n + hk_{b1})$$
$$k_{b4} = f_4(x_n + hk_{a3}, y_n + hk_{b3})$$

where h is the calculated step time between n^th and $(n+1)^th$ time step.

B. Modelling of DC Motor

The back emf e is proportional to the angular velocity of the shaft by a constant factor K_e

$$e = K_e\dot{\theta}$$

In SI units, the motor torque and back emf constants are equal i.e. K_t = K_e. Hence we use K to represent both motor torque constant and back emf constant. We can derive the following equations based on Newton's 2^{nd} law and Kirchoff's voltage law.

$$J\ddot{\theta} + b\dot{\theta} = Ki$$
$$L\frac{di}{dt} + Ri = V - K\dot{\theta}$$

Upon further calculations we can express the governing equations in state space form by choosing the rotational speed and electric current as the state variables. Again the armature voltage is treated as the input and the rotational speed is chosen as the output.

$$\frac{d}{dt}\begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix}\begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} \dot{\theta} \\ i \end{bmatrix}$$

C. Simulation via Hardware in Loop Model

The final model of our project with the buck converter based DC motor is as in the schematic diagram. The fundamental idea behind the schematic is that at the nth instant of time the values of Speed, Armature current and Electrical torque are fed back making the system closed loop. The speed is then compared with the reference value (from the lookup table) of speed at that instant of time and a suitable correction is introduced by the PI controller that tries to mitigate the error. A comparator is put where the signal will only be produced if the value of the modulated PI signal is greater than or equals the sawtooth signal hence removing the noise in the system. Accordingly the a pulse is generated so as to toggle the MOSFET switches ON and OFF. This kind of pulse generation is called Pulse Width Generation (PWM). This circuit in turn generates a specific current and voltage which will generate a closer value of speed to that of reference values. The values of Reference and Actual Speeds, generated PWM signal, Armature voltage/ current, Input voltage/ current, electrical torque are fed into the scope for monitoring purposes. This

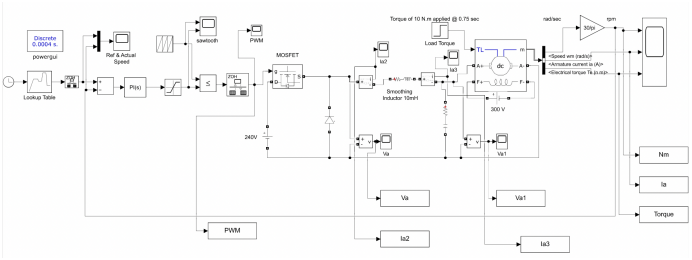

Fig. 4. Buck Converter based DC Motor Drive

model is the digital model of the working of the a Buck Converter based DC motor. This model is in lieu of the unavailability of the hardware to implement this approach. In the further sections we will observe the recognition and classification of various attacks on this model using multiple Machine Learning Models.

III. Hardware in Loop Based Simulation and Visualization of Attacks

The model was implemented on Typhoon HIL software for HIL simulation and data was gathered accordingly. The waveforms of the model under normal condition and under multiple attacks is as show below.For our case study we have considered five different types of attacks on the controller:

- Attack on the speed sensor
- Slow attack on controller
- Change in Reference values
- Change in the parameter of controller
- Multiple attacks

The wave-forms in case of normal and multiple attacks scenarios are perceived in the following figures.

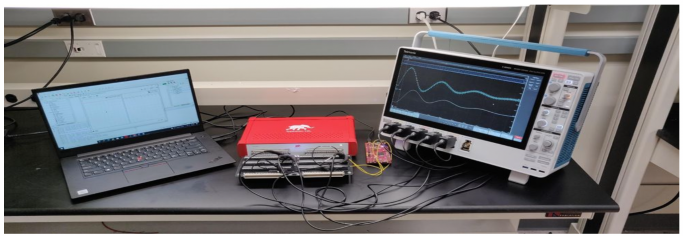

Fig. 5. Setup for Data Collection via HIL Model

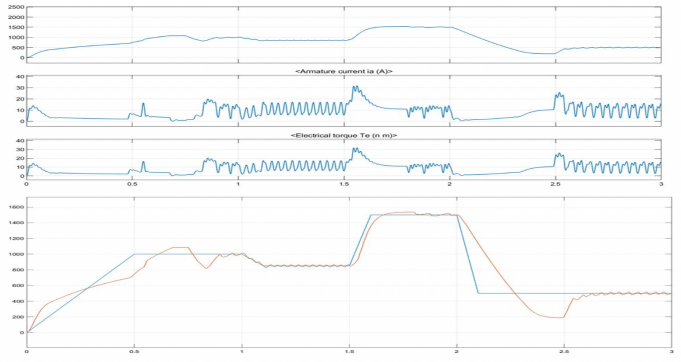

Fig. 6. Waveforms for Model Under Normal Conditions

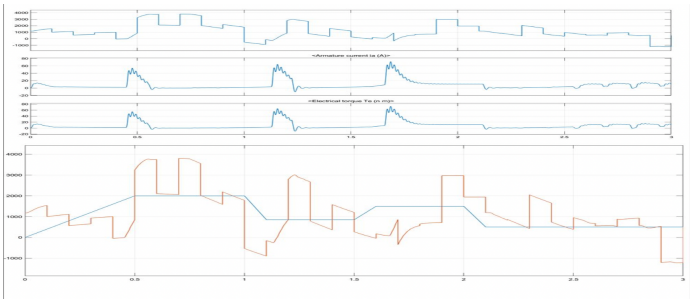

Fig. 7. Waveforms for Model Under Multiple Attacks

IV. Machine Learning Algorithms to Extract Patterns and their Classification

In order to respond to the risks discussed above, we propose to investigate the features of various kinds of the signals resulting from the various scenarios mentioned above. After the feature extraction, we train various Machine Learning models to identify and classify them into various kinds of threats.

In each case, we perform one technique to process the signal and extract features. We then passed the signal through various ML based Classification techniques, and compared the results of training. Finally we pass a faulty test signal to evaluate the performance and accuracy of the classifier model. For our case studies, we have tested with three techniques:

- *Averaging:* We slide a 'window' over the signal that captures a small subset of the signal, and computes its mean at each segment. This makes the signal smooth. Post running a variety of training algorithms, we got the

best results on the resultant signal with the Fine Tree model. This technique uses Decision Trees where each node is a Neural Network. The Fine tree has many leaves to make finer distinctions between the different classes.

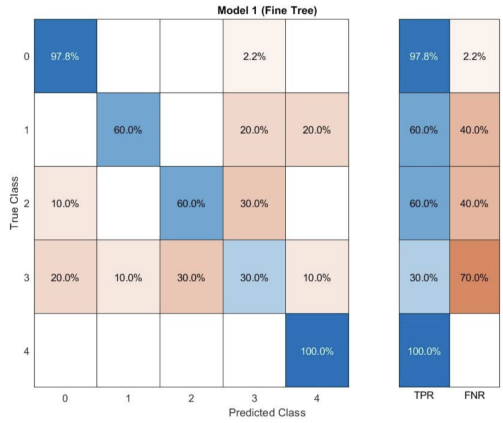

Fig. 8. Confusion Matrix for Averaging

We can see we have achieved reliable identification from the confusion matrix. On proceeding to test this classifier on a test signal we obtain satisfactory results; the classifier can detect class 1, 2, 3 and 4 attacks with reasonable accuracy. In addition it can detect the simultaneous cases when Type 2 and type 3 or type 1 and Type 4 attacks occur simultaneously.

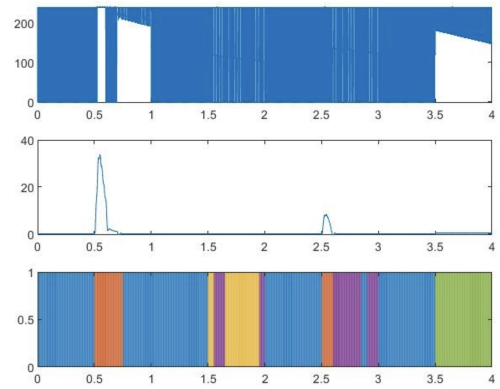

Fig. 9. Result of attack detection and classification for Averaging

- *Slope:* We record the varying gradient of the signal at each point to extract the features. We train various models on detecting these features from the training signal to obtain the confusion matrix for each algorithm. Base on our tests we found the Random Under-sampled Boosted Trees model to offer the best results.
  RUBoost Trees can efficiently handle class imbalance issues, in addition to the high dimensional effectiveness advantages of Support Vector Machines. On testing this classifier model against our test dataset, we can see that the RUBoosted trees works very effectively in capturing

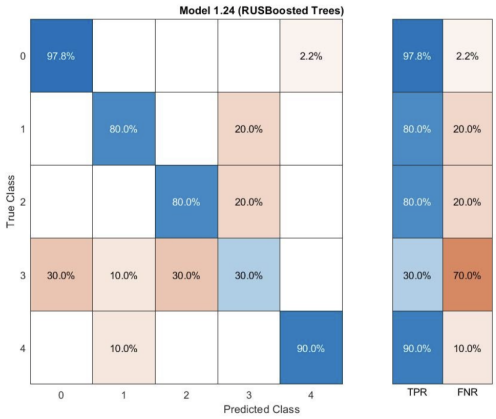

Fig. 10. Confusion Matrix for Slope based Classification

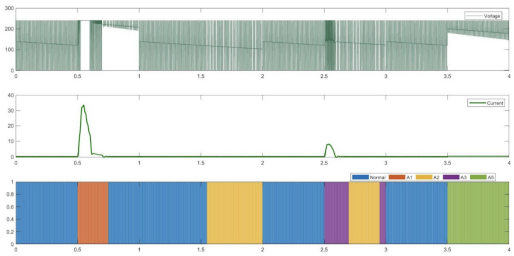

Fig. 11. Result of attack detection and classification for Slope

the problems in the signals at the right timestamp, with negligible delay.

- *Skewness:* Skew is a great identifier of variance in a set of data, and hence makes a great case to use in feature detection and wavelet classification. Multiple training algorithms were used for training the data on the skews. We found Tri-layered Neural Network (TNN) and Bagged Trees methods to offer the best results.

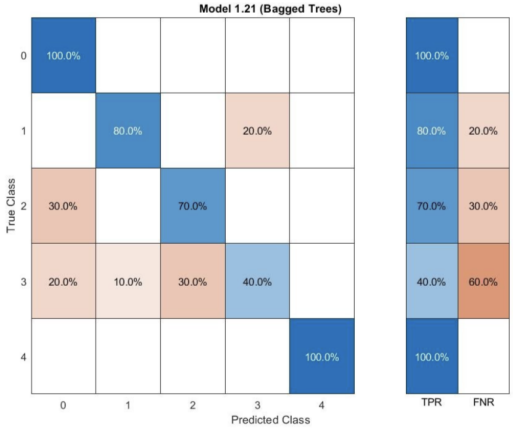

Fig. 12. Confusion Matrix for Skewness Based Classification

The strengths of using Skew as a training feature is the ability to identify multiple attacks concurrently. A major limitation of using Skew as a feature is that most classifier models will falsely misidentify class 4 attacks as class1.

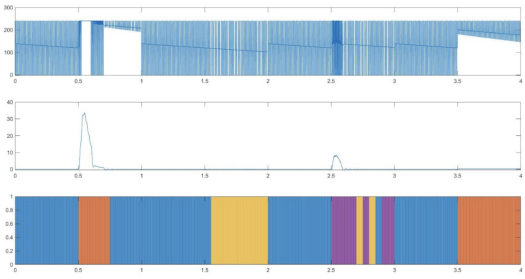

Fig. 13. Confusion Matrix for Skewness Based Classification

V. Conclusion

With this approach of developing a digital twin of the system can rightly ascertain the malicious signals that the system comes across during runtime. In this way any infringement will be filtered out by the digital twin and will not allow any kind of malware to take over the system. This in turn also leads to safety of the driver with no compromise at any end. The model was validated using Hardware in Loop and multiple cases of 2^{nd} generation cyber attacks i.e. Hardware attacks are taken into consideration. The detection of attacks is based on the comparison of the actual system with the clean slate i.e. Digital twin and accordingly the error is fed to multiple Artificial Neural Network (ANN) algorithms for classification. The study of the comparisons of various classification models based on different features is observed and plotted in the confusion matrix. The limitations of this model include no real time infringement detection that can possibly lock the digital twin out of the loop hence causing catastrophic systemic failure. Given both the digital twin and the physical model use the same reference values, any kind of breach at that end would not be detected. Future goals would include the necessary corrections for the above limitations that would increase the reliability of the system and the deployment of this model on a hardware setup. Further improvement in features would include the deployment of the digital twin on a cloud platform like Amazon Web Services (AWS) or Azure.

References

[1] I. Sajjad, D. D. Dunn, R. Sharma, and R. Gerdes in Proceedings of the First ACM Workshop on Cyber-Physical SystemsSecurity and/or PrivaCy. ACM, 2015, pp. 43–53.

[2] R. M. Gerdes, C. Winstead, and K. Heaslip in Proceedings of the 29th Annual Computer Security Applications Conference. ACM, 2013, pp. 99–108.

[3] Y. Fraiji, L. B. Azzouz, W. Trojet, and L. A. Saidane in Wireless Communications and Networking Conference (WCNC), 2018 IEEE. IEEE, 2018, pp. 1–6.

[4] K. Harnett, B. Harris, D. Chin, G. Watson et al. in John A. Volpe National Transportation Systems Center (US),Tech. Rep., 2018.

[5] S. Sripad, S. Kulandaivel, V. Pande, V. Sekar, and V. Viswanathan in arXiv preprint arXiv:1711.04822, 2017.

[6] F. Li, Y. Shi, A. Shinde, J. Ye, and W. Song in IEEE Internet of Things Journal, accepted.

[7] B. Chen, S. Mashayekh, K. L. Butler-Purry, and D. Kundur in Power and Energy Society General Meeting (PES), 2013 IEEE. IEEE, 2013, pp. 1–5.

[8] D. Kundur, X. Feng, S. Liu, T. Zourntos, and K. L. Butler-Purry in Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on. IEEE, 2010, pp. 244–249.

[9] S. Xin, Q. Guo, H. Sun, B. Zhang, J. Wang, and C. Chen in IEEE Transactions on Smart Grid, vol. 6, no. 5, pp.2375–2385, 2015.

[10] S. Liu, S. Mashayekh, D. Kundur, T. Zourntos, and K. Butler-Purry in IEEE Transactions on Emerging Topics in Computing, vol. 1, no. 2, pp.273–285, 2013.

[11] Yang, L. Guo, F. Li, J. Ye and W. Song in 2019 IEEE Transportation Electrification Conference and Expo (ITEC), 2019, pp. 1-6, doi: 10.1109/ITEC.2019.8790574.

[12] M. Farsi, A. Daneshkhah, A. Hosseinian-Far, and H. Jahankhani in Digital Twin Technologies and Smart Cities. Springer, 2020

[13] C. Gehrmann and M. Gunnarsson in IEEE Transactions on Industrial Informatics, vol. 16, no. 1, pp. 669-680, Jan. 2020, doi: 10.1109/TII.2019.2938885.

[14] F. Akbarian, E. Fitzgerald and M. Kihl in 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2020, pp. 1-6, doi: 10.23919/SoftCOM50211.2020.9238162.