Cybersecurity in EVs

### **Why EVs require cybersecurity?**

**vulnerable to hacking.**

* Unlike regular cars, every single component of EVs is linked to a **central computer** tasked with ensuring every part of the car speaks to each other.
* **EV chargers** are often located in remote locations and have very little human supervision. According to researchers at [Concordia University](https://www.concordia.ca/news/stories/2022/02/15/electric-vehicle-charging-stations-are-a-new-focus-for-concordia-cybersecurity-researchers.html), the **firmware and applications used in these devices aren’t always up to typical cyber security standards, leaving EV cars exposed to everything from**[**malware**](https://terranovasecurity.com/what-is-malware/)**to complete takeover by hackers.**

**Possible Attacks:**

**Public Chargers**

* EV chargers — whether located in personal homes or in public areas — typically collect information about a **vehicle's charge rate, identification numbers and drivers' online account information**
* Chargers also [typically connect to a system](https://umdearborn.edu/news/ev-charging-stations-could-be-target-hackers) inside cars known as the **controller area network**, which allows a car's various on-board **electronic components and controllers to communicate** with one another
* This can leave **EVs vulnerable to malware** or maliciously modified chargers altered to damage cars by using the incorrect voltage.

**Phishing Attacks**

Keyless Entry(RFID chips and Apps)

Most EVs have stopped relying on a physical key to access the vehicle, opting instead for apps and RFID chips to unlock the car. Many carmaker apps have sadly had vulnerabilities in that regard, and RFID chips can be relatively easy to clone if a hacker can get their hands on them.

**Prevention**

**Anomaly Detection:**

Using **machine learning to develop a fingerprint of what normal computing activity looks like** when a vehicle is charging, so the system can then identify when something appears out of the ordinary

**From IEEE Explore Library**

The researchers developed a **static authentication process and a secured protocol to generate the smart key** for the user to unlock the vehicle. A **continuous authentication system based on fingerprint, NFC, and facial information is used to authenticate the driver**. Analysis has proven that the proposed protocol for key establishment is secure against popular attacks. In addition, for analyzing the proposed continuous authentication mechanism, the **researchers built a prototype incorporating Raspberry PI as a replica of the car’s computer interfaced with the fingerprint and NFC modules and an Android app that facilitates each factor of authentication.** The preliminary experiment in real-world settings suggests the efficacy of the proposed design.

* **Use Strong Passwords**
* **Software up to date:**

Regular software updates recommended by manufacturer. Never execute 3rd party software of any kind on your car’s computer

**Intelligent Traffic Environment:**

Relies on real-time data from connected road infrastructure and predictive analytics to effectively coordinate traffic across city arteries.

Such traffic management software, coupled with wireless urban connectivity, acts as a backbone for the implementation of an [intelligent transportation management system](https://intellias.com/implementing-intelligent-transportation-system/).

**Simulations**

**OPAL-RT** is a company known for its real-time simulation solutions, and **HIL(hardware in the loop)** refers to a testing methodology where a physical system is connected to a simulation environment.

**Jeep Cherokee Incident**

**In 2015, cybersecurity researchers Charlie Miller and Chris Valasek remotely hacked a Jeep Cherokee through Wi-Fi connection**, leading to the recall of 1.4 million Fiat Chrysler vehicles.3 In 2016, the same pair hacked their Jeep Cherokee again, **requiring physical access through a laptop connected to the OBD II engine diagnostic port.2**

**They were able to disable the transmission and brakes and take over the Jeep's steering, only with the car in reverse, at low speed.**

**The hack worked because the car's steering can be controlled when the car thought it was automatically parallel parking.**

machine learning-based classifier is developed and validated using physics-guided data features in an OPAL-RT hardware-in-the-loop (HIL) simulation testbed.

In power systems, maintaining accurate and timely knowledge of the system states is crucial for effective monitoring, control, and decision-making. Physics-guided deep learning can enhance the accuracy of state estimation by leveraging both data-driven approaches (deep learning) and the fundamental physics governing the power system.

**Vehicular longitudinal dynamics**

It refers to the study and analysis of the motion and behaviour of a vehicle in the longitudinal direction, which is typically along the axis of the vehicle. Longitudinal dynamics primarily involve the motion and forces acting on a vehicle as it accelerates, decelerates, or maintains a constant speed. Understanding vehicular longitudinal dynamics is crucial for designing vehicle control systems, optimizing performance, and ensuring safety

**Prevention Methods:**

* Secure Hardware
* Secure Communication Techniques
* Firewall
* Secure Software Update
* Software based Intrusion detection system

(aims to design a reliable real-time monitoring system )

* physics-guided deep learning approach

(outputs the estimated states by taking

real-time measurements as inputs to neural networks and then

reconstructs measurements considering power system physics This research

will present how to utilize the knowledge about vehicular

longitudinal dynamics and motor drive’s model to improve

the detection accuracy of machine learning. First,)

**Vulnerable parts of EV:**

* Battery Management
* Motor Drives
* Braking
* Steering
* Sensors(3-phase current from the IPM drive)
* Network
* OS
* Sensor (location A: three-phase current from the IPM drive),
* Controller (location C: the output of the PI controller)
* Communication channel with the EMS (location B: torque reference from the higher-level controller).

**List of Data parameters to train AI Model:**

* **Vehicle Speed(We see frequency , magnitude and phase)**
* **Torque reference**
* **Voltage**
* **Current in the electric machine (IPMSM,IM)**

**\*\* Use LSTM architecture (By replacing nodes**

**in the recurrent neural network with memory cells and gating**

**mechanism [58], [59], LSTM can effectively handle long-term**

**dependencies of data.)**

Based on the above hyper parameters, the LSTM model is created by using TensorFlow.

**Structure of basic LSTM Cell**

1. **Fully Connected Layer:**
   * The neural network architecture includes a fully connected layer, also known as a dense layer.
   * In this layer, each neuron is connected to every neuron in the preceding layer.
   * The output of the fully connected layer, denoted as *z*, represents the raw scores or logits associated with each class.
2. **Softmax Layer:**
   * Following the fully connected layer, there is a softmax layer.
   * The softmax function is applied to the raw scores *z* to convert them into probabilities.
   * The softmax function normalizes the scores into a probability distribution, ensuring that the predicted values sum to 1.
   * For a specific neuron �*j*, the softmax activation ��*jy*​ is calculated using the formula: ��=���∑�=1�����*jy*​=∑*i*=1*nc*​*ezi*​*ezj*​​
   * Here, ��*nc* represents the number of target classes, and ��*zi*​ is the raw score of the �*i*-th class.
3. **Classification Output Layer:**
   * The final layer is the classification output layer, which produces the predicted class labels.
   * The class with the highest probability in the softmax output is selected as the predicted class for a given input example.
4. **Interpretation of Softmax Output:**
   * The softmax output �^�*y*^​*j*​ for each class �*j* can be interpreted as the estimated probability that the input example belongs to class �*j*.
   * The class with the highest predicted probability is chosen as the final predicted class label.

Based on the above hyper parameters, the LSTM model is created by using TensorFlow.

Questions

* Check Comments
* D-axis and q-axis current
* Attack Vectors
* LSTM Architecture