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 analysing 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
* Controller (location C: the output of the PI controller)
* Communication channel with the EMS (location B: torque reference from the higher-level controller).
* Attack Vectors: Bluetooth & Cellular
* Wireless interface of the tire pressure monitoring system
* Passive Keyless Entry
* Engine Startup System

**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.

**Limitations of CAN Protocol:**

**Given the CAN protocol limitations, any cryptographic message**

**authentication would have too weak of a key to be useful.**

* **Limited Bandwidth and Resources**
* **Key Distribution Challenges**
* **Timing and Latency Constraints**
* **Compatibility with Existing Systems**
* **Physical Access Challenges**

**CAN PROTOCOL**

* Due to the fact that messages must be broadcast at a high frequency, the encryption/authentication mechanisms may lead to delays.
* Authentication techniques for the in-vehicle network would not necessarily prevent attackers from remotely attacking the car and gaining access using its own networkinterfaces.

This shortcoming is mitigated by machine learning anomaly detection systems. One complication is that the system must monitor not only

the CAN bus traffic, but rather the different vehicle interfaces

(e.g., OS, Network, and CAN).

Dimensionality reduction technique -PCA

How to Detect CAN Bus Anomaly?

* Frequency-Based Technique:
* Message ID has regular Frequency

Lightweight intrusion detection algorithm based on the fact that each message ID has a regular frequency, and when attackers inject messages into the CAN bus, the message frequency changes abruptly.

* Most in Vehicle network messages are periodic:

Most in-vehicle network messages are periodic and broadcast over CAN, and suggested exploiting the time intervals of these periodic messages as ECU fingerprints. These methods are mainly effective for periodic messages, so an attacker who injects messages aperiodically may go undetected.

Moreover, when the ECU is itself the source of the malicious packets’ IDs, attacks may go undetected.

* Statistical Techniques
* construct a “normal” baseline and then to identify deviations from the norm
* Entropy-Based Model

The basic intuition is that due to the clear and restrictive specification of the in-vehicle traffic, the entropy is relatively low, and therefore, attacks (e.g., changing packet payload, packet injection) would cause the entropy to increase.

* Limitations

In order to detect low-volume attacks, one must build an anomaly detector for each message ID. Han et al. [15] divided the data into four categories (engine, fuel, gear, and wheel) and used a one-way ANOVA test to identify abnormal activity.

Machine learning Models Used till date:

**HMM(Hidden Markon Model) using new temporal technique**

* Temporal detection technique involves analysing patterns and behaviours over time to identify deviations from normal or expected behaviour.

**LSTM(Long Short Term Memory)**

**DT(Decision Tree)**

**SVT(Support Vector Machine)**

**KNN(K-Nearest Neighbour)**

**NB(Naïve Bayes Classifier)**

**Gaussian Mixture Model(GMM)**

**IDS based on Deep Neural Network(DNN)** Deep belief Network **(DBN) were used to initialize the DNN parameters as a preprocessing stage.**

**DETECTION TECHNIQUE**

A static threshold is used (either for single observation or for sequences), and scores crossing the threshold are flagged as anomalous. However, a static threshold can be inaccurate in many applications such as temporal data with time and history-sensitive characteristics.

**Hence used Additional regression model:**

* The training set is divided into two parts: *P*1 and *P*2. The first part, *P*1, will be used to train the HMM, while the second part, *P*2, will be used to build a regressor that will predict the log-likelihood for time interval *t*.
* we compare the event log likelihood computed from the HMM model to the regression model predicted log-likelihood.

**Datasets:**

**SUMO (Simulation of Urban Mobility)**

**Events:**

**well-defined occurrence in the vehicle, generated by one of the on-board security systems**

Event types include login, logout, door open, door close, file access, running app, install app, app update, USB inserted, network usage, etc.

Each event contains different attributes.

A file access event contains attributes such as file type (root, protected, and public), action (read, write, and execute), etc.

**Story Definition:**

A sequence of events sent from the vehicle to the backend.

A *drive* is therefore composed of stories.

Examples: Added stories such as installing application, playing music (e.g., from stream, USB, and phone), GPS access, open flows (e.g., weather), open ports, etc.

**Adding Noise**

injected network usage, open flows (weather and GPS applications communicating with servers), connected devices (Bluetooth communication, USB successful/unsuccessful insertions), open/closed ports, different file accesses, drive cancellation (the driver entered the vehicle, but exited before he started driving,

**Data Transformation**

**Events to suitable HMM training data through:**

* **Event ID Transformation: Model is trained based only on the event IDs**
* **Discrete Transformation: Doubt Events transformed into discrete feature vector using configurable buckets**
* **Attack Scenarios: Suspicious activity detected through Missing events, unknown sequence of events, unrealistic order of events known sequence of events with different attribute values**
* **Out of order Events**
* **Out of context**: sudden dec/inc in speed or extensive outbound communication with an unknown IP in an unusual context.
* **USB firmware update attack**: found an exploit in the USB update key exchange mechanism which allows them to take the original USB and insert it, wait for the authentication process to finish, extract the original USB, and replace it with a malicious one.
* **OTA malicious updates**:
* **Malicious application installation**:
* **HMM Model Generation**: Compared different models and chose the vest log-likelihood.
* **Tests**: 2 Main goals:
* Test the developed algorithms and system architecture
* Analyse the dynamic temporal threshold performances
* Experiments conducted in:
* **Full Drives (Offline):** The system waits until the driver has stopped the vehicle and logged out, then it pulls the full sequence of events that occurred during the drive (and were stored in the backend) and tests it against the vehicle’s HMM and regression models.
* **Drive Prefixes (Online):** The system tests each new event as soon as it is sent from the vehicle, and gives an alert when the accumulated prefix of events is identified as anomalous.

**Conclusion:**

Evaluation process considered a comprehensive set of metrics and scenarios, and the regression model with a temporal threshold demonstrated superior performance in terms of anomaly detection. This information can be valuable in making informed decisions about the choice of transformation methods for anomaly detection based on the specific requirements and goals of the application.

**Deployment: 2 Main Approaches:**

Provide trade-off between several aspects including **detection latency, detection scale footprint and bandwidth consumption**

**On-Board Architecture:**

* Will **provide real-time anomaly detection** **due to the fast communication between the data collector and the detection engine.**
* However, this **method is resource-intensive on the edge units.**
* Furthermore, it is **far more challenging to serve car fleets and detect cross-fleet anomalies using this architecture.**

**Backend Architecture:**

* On the other hand, in the backend architecture, **serving car fleets and detecting cross fleet anomalies is much** **easier**
* There is **a certain latency in the data transmitted from vehicles to the cloud**. Moreover,
* This approach requires **enormous bandwidth.**

**Challenges**:

* To apply real-time anomaly detection using an HMM, overcoming the low computational power and limited memory resources.
* To detect anomalies in real-time for a car fleet.

**Decision**:

To address these challenges, we suggest a **hybrid platform with an HMM anomaly detection mechanism:**

**By moving the detection mechanism to the cloud,** we can overcome the low computational power and limited memory resources of the vehicle.

However**, transmitting raw CAN/OS/Network packets to the cloud will require enormous bandwidth and a high transmission rate, perhaps overwhelming the backend**. Therefore, we **suggest using a light-weight component** that could be integrated into the vehicle that can extract important data based on configurable rules, and transmit it back to the backend with all the selected features, which means our data will not be raw packets; rather, it would be at a higher level of abstraction and include only relevant data extracted from applications, network traffic, chosen sensors, CAN bus, etc.

The backend would store HMMs learned for a large number of vehicles. **By using a cloud-based platform, we can serve car fleets and possibly apply advanced analytics for identifying correlations between vehicles and identify in-progress attacks on vehicles using anomalies detected from other vehicles.**

Using this approach, we **can either train an HMM for each car and store it in a distributed database**, or **build a model for groups of cars with common characteristics using clustering-based techniques.**

**CONCLUSION OF ADVANCED ANALYTICS:**

**Temporal features** are time domain features that capture recurring patterns or sequences of visual elements over time. approach handling the vehicle’s collected data and a new temporal-based detection technique

**Regression model based on temporal features** (e.g., time since drive start, consecutive arrival times, etc

System monitors not only the CAN bus, but also **monitors different vehicle interfaces (OS, Network, and CAN)**, and was proven to be highly capable of detecting real-life com plicated anomalies which involve a wide number of features from different interfaces

**Capable of** **monitoring car fleets.** Furthermore, it allows us to train HMMs either for individuals or for groups of cars with common characteristics using a clustering preprocessing stage.

**Machine learning for enhancing transportation security: A comprehensive**

**analysis of electric and flying vehicle systems**

Focused on a cost-efficient and secure vehicular fog cloud computing (VFCN) system.

**Cost-efficient and secure VFCN**:

This indicates the research focuses on creating a VFCN system that optimizes both cost and security aspects.

**Mobility-aware multi-scenario offloading phase (MAMSOP):**

This suggests the system incorporates a mechanism to handle the dynamic mobility of vehicles and offload tasks efficiently based on different scenarios encountered.

**Role of ML in enhancing security**

Various **CNN’s** used

* **Secure vehicular fog cloud computing** (VFCN) comprising **a Mobility-aware multi-scenario offloading phase** (MAMSOP)
* **Full Homographic encryption-based scheme**
* **Fully Polynomial-time approximation scheme** (FPTAS)
* **Homomorphic federated learning-enabled pedestrian**  **and vehicle detection system (HMFLS)**

**Advantages of HMFLS**

* **Security**
* **Efficiency**
* **Integration capabilities.**
* System **utilized fog nodes** and **cloud servers to collect real-time data** and **train** ML models for pedestrian and vehicle detection.
* Incorporated **homomorphic encryption**, enabling computations on encrypted data without decryption, **ensuring privacy and confidentiality.**
* **Generative Adversarial Networks**
* **VGG19 to train pedestrian and vehicle images and extract features**
* **Android-based application**
* **The challenges and issues faced in sustainable transport applications, such as battery power consumption and execution accuracy.**
* **Fuzzy-based energy-efficient decision support system (FBEES)**

**Minimize**

* **energy consumption**
* **delay**
* **cost while**

**Enhancing**

* **Scheduling accuracy**

Questions

* Check Comments
* D-axis and q-axis current
* Attack Vectors
* LSTM Architecture
* Cellular Attack to overtake over vehicle
* Difference between DSRC & RFID & NFD