Securing Smart Vehicles from Relay Attacks Using Machine Learning

Usman Ahmad*, Hong Song*, Awais Bilal*, Mamoun Alazab† and Alireza Jolfaei‡

*School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081, China

Email: usmanahmad@bit.edu.cn, anniesun@bit.edu.cn, awaisbilal@bit.edu.cn

†Charles Darwin University, Darwin, Australia, Email: mamoun.alazab@cdu.edu.au

‡Macquarie University, Sydney, Australia, Email: alireza.jolfaei@mq.edu.au

Abstract

Due to the rapid developments in intelligent transportation systems, modern vehicles have turned into intelligent transportation means which are able to exchange data through various communication protocols. Today's vehicles portray best example of a Cyber Physical System (CPS) because of their integration of computational components and physical systems. As the IoT and data remain intrinsically linked together, the evolving nature of the transportation network comes with a risk of virtual vehicle hijacking. In this paper, we propose a combination of machine learning techniques to mitigate the relay attacks on Passive Keyless Entry and Start (PKES) systems. The proposed algorithm uses a set of key fob features that accurately profiles the PKES system and a set of driving features to identify the driver. First relay attack detection is performed, and if a relay attack is not detected, the vehicle is unlocked and algorithm proceeds to gain the driving features and use neural networks to identify if the current driver is whom he/she claims to be. To assess the machine learning model, we compared the Decision Tree, SVM, and KNN method using a three-month log of a PKES system. Our test results confirm the effectiveness of the proposed method in recognizing relayed messages. The proposed methods achieves 99.8% accuracy rate. We used a Long Short-Term Memory (LSTM) recurrent neural network for driver identification based on the real-world driving data, which is collected from a driver who drives the vehicles on several routes in real-world traffic conditions.

Index Terms

Machine learning, neural networks, PKES, relay attacks, driver identification, security.

I. Introduction

A Cyber Physical System (CPS) refers to tightly coupled computational components and physical systems, enabling humans with advanced capabilities to manage and interact with physical world. CPS application areas are not only limited to the power, industry and healthcare, but have also include automobile sector. Today, most of the research that is being conducted in automotive domain is in electronics and software, rather than in mechanical engineering. Therefore, automotive cyber physical systems have transformed cars from electro-mechanical units into advanced and modern means of transport offering safety, security, efficiency and convenience. Automotive cyber physical systems in modern high-end cars are complex, hosting multiple Electronic Control Units (ECU),

millions of lines of code and internal networks connecting sensors with control units and actuators. However, this correlation between cyber and physical world also poses new security challenges. Since attacks in cyber world can have devastating effects in the physical world, ensuring security in automotive cyber physical systems is important to provide availability, reliability and dependability.

Modern vehicles contain new connected technologies that aim to provide improved fuel economy, safety features, and seamless connected experience. Despite the benefits of such technologies, vehicle's increasing autonomy and connectivity carries new cyber threats. Recently, the national highway traffic safety administration, and the department of transportation, and the Federal Bureau of Investigation (FBI) jointly released a report about the increasing graph of vulnerabilities in the modern vehicles with respect to remote exploits [1]. Locking the car doors, disabling the brake system, and shutting down the engine are examples of vehicles security attacks [2]. The hackers only need to be within the proximity range of the vehicle's communications to implement their attacks.

Waraksa et al. introduced the concept of PKES [3] and the algorithm was first used by Mercedes-Benz in 1998 [4]. Since then, many car manufacturers have employed similar PK systems. Despite the comfort of PKES, it is prone to man-in-the-middle attacks [5], [6]. People, who have the PKES system in their vehicles, have started keeping their key fobs in faraday cages or freezers to protect it from relay attacks. The thieves use inexpensive power amplifiers (which could cost less than 15 USD) to relay the signals and break into the vehicles. Fig. 1 shows such attacks. The main reason for the success of relay attacks is that PKES systems only verify the proximity of key fob rather than the physical location of the key fob.

A large number of cryptographic and authentication techniques have been proposed to encrypt the communication between the key fob and vehicle to authenticate the key fob. However, the PKES system is still vulnerable to relay attack because the attackers do not need to decrypt or modify the communication. In fact, they just boost the communication signal between the vehicle and key fob to ultra-high frequency. Moreover, computationally strong encryption or decryption can be very resource and time consuming [7], [8], which is an important concern in a real-time transportation system [9]. The past literature also shows other solutions such as distance bounding protocols [10] and location-based authentication mechanism [11]. The location-based authentication mechanism does not protect against the relay attack, because the attackers can still relay the real key fob data.

This paper proposes a double security layer. firstly, we protect the PKES system from well-known relay attacks using key fob features. Secondly, we complement the method by adding a driver identification mechanism which uses driving behavior/features. We extract a set of key fob features and driving features. Key fob features include time, date, location, type of day, elapsed time, key fob acceleration, and signal strength. Driving features include accelerator pedal positions, brake pedal positions, and following distance. The proposed security layer employs machine learning techniques to identify whether a key fob signal follows the pattern of relayed signals. To improve the rate of detection accuracy we used 10-fold cross-validation. We compared the performance of three well known algorithms of machine learning, including the Decision Tree, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) with respect to training time, prediction time, accuracy rate, and confusion matrix. We tested our method using a three-month log of a PKES system. If a relay attack is not detected, the vehicle is unlocked, and the algorithm proceeds to gain the driving features. The algorithm uses neural networks to identify if the current

Fig. 1: Relay attack on a PKES system

driver is who they claim to be. We used the LSTM recurrent neural network for driver identification based on the real-world driving data which is collected from a driver who drives the vehicles on 12 routes in real-world traffic conditions. Fig. 2 demonstrates the block diagram of our proposed solution. Our driver identification solution even works, if a robber snatches the car by threat of force or terrifying the driver.

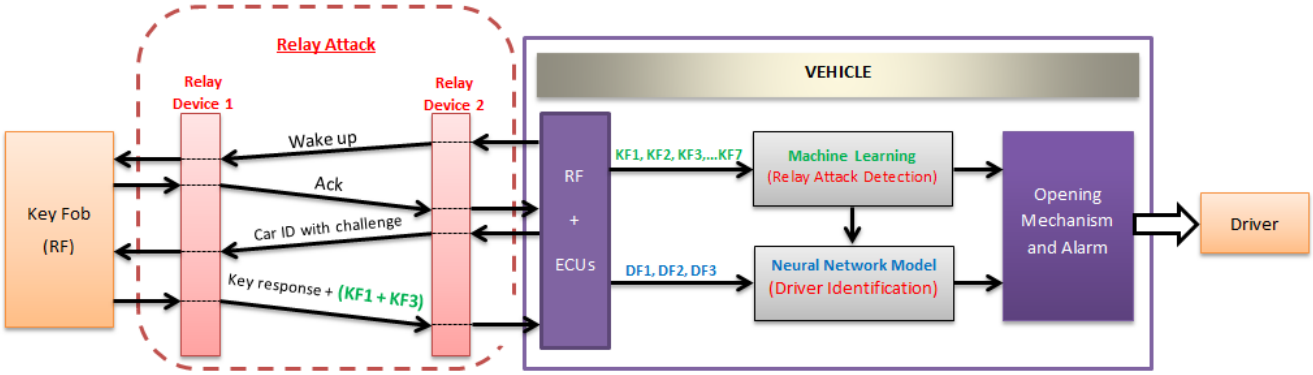

Fig. 2: Block diagram of the proposed solution

The remainder of the paper is organized as follows: Section II reviews the related work. Section III describes the methodology of the proposed solution. Section IV assesses our solution and shows the comparative analysis of machine learning and neural network model. Section V illustrates the experimental setup using real-world data. Finally, Section VI concludes the paper.

II. Related Work

The recent increase in thefts of smart vehicles [12] has raised demands for more secure PKES systems. Vehicle manufacturers and insurances companies require secure systems to decrease theft rates. In [13], Miller and Valasek highlighted several security vulnerabilities [14], and as a proof of concept, they performed various remote attacks against the 2014 Jeep Cherokee. In [15], [16], the authors demonstrated the relay attack on a PKES system by amplifying the low-frequency radio signals between the key fob and vehicle.

In [10], the authors proposed the distance bounding protocols, which calculate the distance between the key fob and vehicle by measuring the round-trip and the time-of-flight of the signal. Asmar et al. [11] developed a location-based authentication mechanism for the PKES system, which continuously determines the location of the key fob after the vehicle is unlocked and on the move. But this solution does not protect against the relay attack, because the attackers can still relay the real key fob data. In [17], Guan et al. used hardware-based encryption method transactional memory to encrypt the communication. Guan et al.'s method continuously monitors and protects the encryption key [18] from any malicious program that tries to access it. In [19], [11], [20] and [21], to prevent man-in-the-middle attacks, the use of cryptographic solutions were suggested. However, computationally strong encryption or decryption can be very resource and time consuming, which is an important concern in a real-time transportation system [22], [23], [24].

The study of the literature shows that machine learning and deep learning have extensively been used for securing the key fob communications. In [25], the author used the fuzzy-based machine learning system to learn the good and bad signal behavior. In [26], the author used an active machine learning approach to improve the performance by reducing the false positive rates for on-road vehicle recognition and tracking. In [27], the authors proposed a framework using machine learning based on a set of security features that accurately profile the PKES system and identify irregular PKES behavior. The Use of the log of IoT devices as a dataset for the implementation of machine learning and neural networks and for prediction is a useful method.

In [28], the authors proposed a driver identification algorithm based on the Principal Component Analysis (PCA) approach, using lateral and longitudinal accelerations. In [29], the authors proposed a solution for real-time driver identification using a combination of unsupervised anomaly detection and neural networks. In [30], the authors developed an online status-aware solution for driver identification based on unlabeled data named SafeDrive.

III. Methodology

The proposed solution can be divided into two consecutive stages, as shown in Fig. 3: relay attack detection on a PKES system based on supervised machine learning model and driving behavior identification using neural networks. Firstly, relay attack on PKES is detected on the seven key fob security features (KF) including time, date, location, type of day, elapsed time, key fob acceleration, and signal strength, to detect irregular key fob behavior. If a relay attack is detected, the algorithm doesn't unlock the vehicle and generate an alarm, else, proceeds to driver identification. The neural network model is then built to identify the driver's behavior using three driving features (DF): acceleration, brake, and following distance.

The Classification and Regression Trees (CART) is a powerful machine learning model for both classification and regression [31]. We used an optimized version of the CART algorithm using key fob security features to detect the abnormal behavior of key fob. We have used the LSTM recurrent neural network, a variance of neural networks introduced by Hochreiter et al. [32], for driver identification. We choose a recurrent neural network because it performs better on sequential data as we have a sequential dataset of acceleration, brake, and following distance.

TABLE I: Key Fob Features (KF)

Feature	**Description**
Location	Location of key fob when transmitting the unlock signal
Time	Time at which the key fob sends the unlock signal
Date	Date at which the key fob sends the unlock signal
Type of day	Indicates weekend or working day
Elapsed time	Elapsed time since the occurrence of last unlock action
Signal strength	Strength of the key fob signal
Key fob acceleration	Indicates that the driver is moving while unlock signal is transmitted

TABLE II: Driving Features (DF)

Feature	**Description**
Acceleration	Change in accelerator pedals position (0 to 10 level)
Breaking	Change in brake pedals position (0 to 10 level)
Following distance	Following distance from the very next vehicle (meters)

A. Security Features

We extract two types of security features: key fob features and driving features summarized in Table I and II, respectively.

1) Key Fob Security Features: Two key fob security features: key fob acceleration and location are embedded in the PKES response message, and they are transmitted with the unlock code. The rest of the five features including time, date, type of day, elapsed time, and signal strength are obtained from vehicle's ECU. The key fob security features used in profiling the normal behavior are outlined below:

Key fob acceleration (KF_1). It is the most significant attribute to detect the relay attack. For example, if the key fob is on the office table, then its acceleration rate must be zero. In such a case, the vehicle must not be unlocked. If the key fob acceleration rate was not zero, it would imply the driver is moving towards the vehicle. We define $KF_1 = \{1, 0\}$ as the key fob is accelerating or not, respectively.

Signal strength (KF_2). This feature represents the strength of the key fob signal. A significant change in the signal strength could potentially indicate relay attacks because the signal transmitted by the relay device would normally have a higher or lower signal strength compared to the original key fob signal that is transmitted from a proximity range. Thus, the signal strength must be within a certain range, and any out-of-range deviation could point to potential relay attacks. We represent the signal strength using the set $\{0, 1, 2\}$, where 0, 1, and 2 denote low, normal, and strong, respectively.

Location (KF_3). This attribute determines the current location of the key fob while transmitting the unlock

signal. This attribute ensures that the key fob is near the vehicle. We define $KF_3 = \{0, 1\}$, where 1 and 0 denote the key fob is near the vehicle or not, respectively.

Time (KF_4)*, date* (KF_5) and *day of unlock action* (KF_6). These attributes state the time, the date, and the day at which the unlock signal is transmitted from the key fob. The time of unlocking can be at any time. When we add the feature of the day with the time, it becomes more meaningful. This is mainly because the time of unlocking is strongly correlated with the working hours of the driver. For example, the unlocking habits are normally different on weekdays and weekends. Moreover, the attribute of date will enhance the learning performance during the holidays. For the time KF_4, 24 values are labeled for representing hours as $\{1, 2, 3, ..., 23, 24\}$. For the date KF_5, there are two categories $\{1, 0\}$ which denotes holiday and workday. For the type of day KF_6, we have seven days in a week $\{1, 2, 3, 4, 5, 6, 7\}$ as Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, respectively.

Elapsed time since the occurrence of last action (KF_7). This feature states the time passed since the happening of an unlock action and determines the frequency of the action. It is used to assess if there is any significant change in the regularity of an unlock action. For the elapsed time KF_7, we classify it into four categories $\{1, 2, 3, 4\}$ which denote less than 8 hours (office timing), less than a day, less than three days, and less than one week, respectively.

2) Driving Security Features: Classification of driving behaviors based on driving features has been studied using various statistical and computing models [29], [30]. Automatic driver's behavior can be detected whether a vehicle has been attacked. A change in accelerator pedals positions, brake and steering wheel control, indicates a shift in driving patterns which deviate from normal driving behavior. Different vehicle manufacturers use differing CAN-BUS models and protocols, and thus, it is not easy to collect the driving features data. Hence, we construct our model based on such attributes that can be collected from almost any kind of vehicles and they are directly influenced by driving behaviors. We extract three driving features: acceleration, brake, and following distance, as listed in Table II. Acceleration and brake pedal positions are digitized to 0 to 10 levels, such that 10 corresponded to full acceleration or completely braked position. The distance from the very next vehicle is calculated in meters.

B. CART Algorithm

The CART algorithm constructs binary trees using the feature and thresholds that yield the largest information gain at each node. The information gain is based on the decrease in entropy after a dataset is split on a feature. For a training vector $X = [x_1, x_2, \ldots, x_n]^t$ and a label vector $Y = [y_1, y_2, \ldots, y_l]^t$, where for $(1 \leq i \leq n)$ and $(1 \leq j \leq l)$, x_i and y_j belong to $\mathbb{R}$. Given training vectors and their corresponding label vectors, a decision tree recursively partitions the space, such that samples with the same labels are grouped together. Let Q denote the data at node m. For each candidate split $\theta = (j, t_m)$, where j represent a feature and t_m denotes a threshold, the data is partitioned into $Q_{left}(\theta)$ and $Q_{right}(\theta)$ subsets as below:

$$Q_{left}(\theta) = \{(x, y) | x_j \leq t_m, y \in \mathcal{R}^l\}, \tag{1}$$

$$Q_{right}(\theta) = Q \setminus Q_{left}(\theta). \tag{2}$$

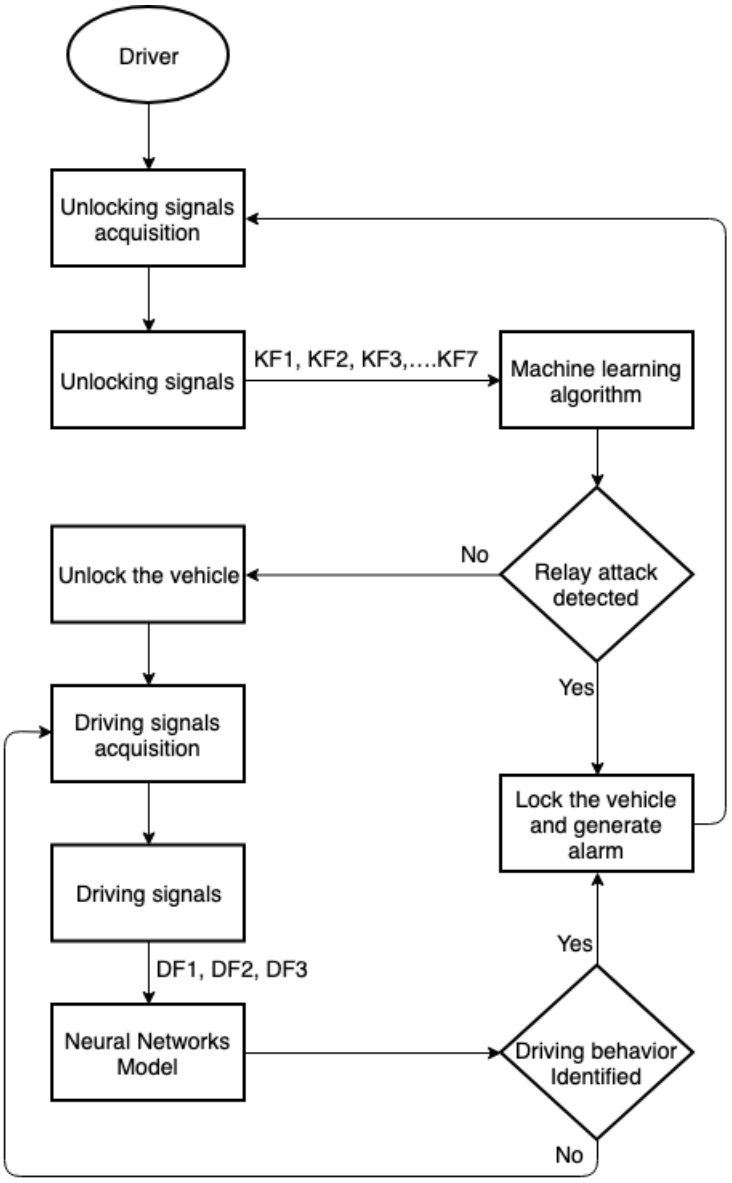

Fig. 3: Flow chart of proposed solution

The impurity of node m is computed using the impurity function $H(\cdot)$, the choice of which depends on the task being solved (classification or regression) and in our case it is classification

$$G(Q,\theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta)). \tag{3}$$

To select the parameters that minimize the impurity

$$\theta^* = \text{argmin}_\theta \, G(Q,\theta). \tag{4}$$

The subsets $Q_{left}(\theta^*)$ and $Q_{right}(\theta^*)$ are recursed until the maximum allowable depth is reached, that is, $N_m < \min_{samples}$ or $N_m = 1$.

Node m denotes a region R_m with N_m observations. In node m, the proportion of training observations in the m-th region that are from the k-th class is

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i), \tag{5}$$

where $I(y_i = k) = 1$ and $I(y_i \neq k) = 0$.

The impurity function $H(X_m)$ is a measure of total variance across the K classes, defined as

$$H(X_m) = \sum_{k=1}^{K} p_{mk}(1 - p_{mk}), \tag{6}$$

where X_m is the training data in node m. A small value of $H(X_m)$ shows that node m contains predominantly observations from a single class.

The cross entropy is given by

$$H(X_m) = -\sum_{k=1}^{K} p_{mk} \log(p_{mk}). \tag{7}$$

The misclassification rate in node m is computed as

$$E(X_m) = 1 - \max_m \{p_{mk}\}, \tag{8}$$

where X_m is the training data in node m.

IV. Experiments and Evaluation

We used the Python programming language for machine learning and neural networks implementations. The Decision tree is designed and trained by using Scikit learn based machine learning library. The LSTM recurrent neural network is designed and trained in Keras, a Tensorflow based deep learning library. Decision tree graph is generated on the dataset of key fob features, shown in Fig. 4. We compared the decision tree algorithm with well-known machine learning algorithms, including SVM and KNN with respect to accuracy rate, confusion matrix, training time, and prediction time. We used k-fold cross-validation for generating independent datasets to evaluate the decision tree results.

A. Data Collection

The observed features can be categorized into two groups: 1) key fob features: $KF_1, KF_2, KF_3, \ldots, KF_7$; 2) driving features: DF_1, DF_2, and DF_3. The features were collected using the Toyota Prius. The key fob dataset contains 500 records which are derived from a three-month log of PKES system. The driving dataset consists of 12 driving trips of 5 minutes from a driver where each trip comprises approximately 300 records, so the total observations are 3600. We have preprocessed the driving dataset to create time-series of input data that represent driving behaviors over 1 sec. to enhance the classification capability. The line graph of an example of the accelerator and brake pedal position of a 5-min (300 seconds) vehicle's trip is shown in Fig. 5.

B. Machine Learning Results

We used 80% of the key fob dataset for training and the rest of it for testing. Fig. 6 shows a comparison of the relay attack detection accuracy rate of the decision tree, SVM, and KNN.

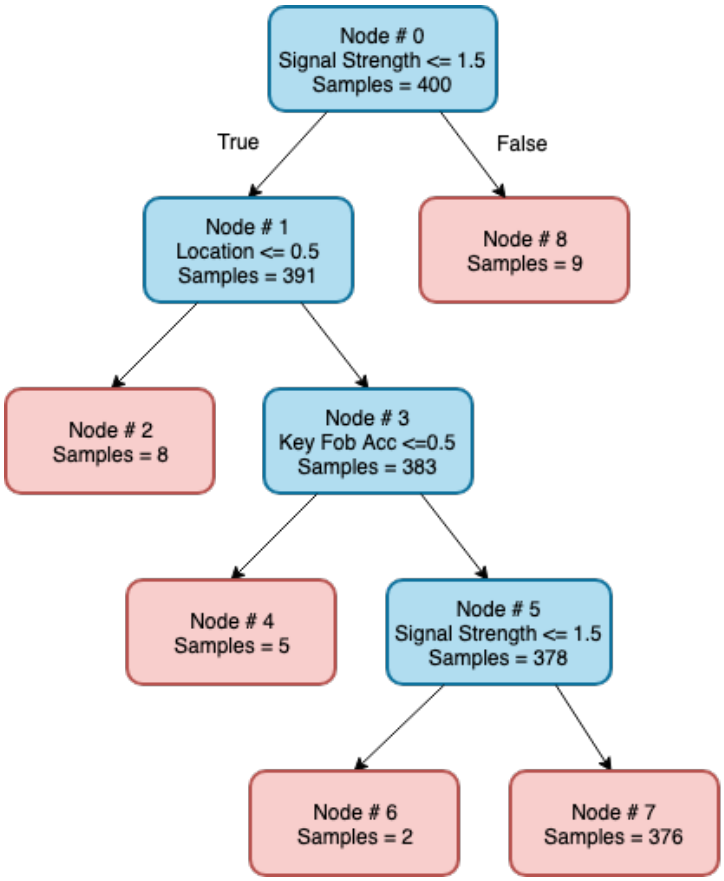

Fig. 4: Decision Tree Graph

1) Train/Test Split: Fig. 6 compares the detection accuracy rates of the decision tree with that of SVM and KNN algorithms. The test results show that the decision tree has the best performance compared to SVM and KNN algorithms. The decision tree and KNN have same and comparatively fast training speed among all that is, 1 ms. The SVM demonstrates a slower training speed, that is, 9 ms. The decision tree and SVM have the fast prediction speed, that is, 1 ms and 0.5 ms respectively, where KNN has 2 ms.

2) K-Fold Cross-Validation: We used a 10-fold cross-validation to overcome the limitations of the train/test split procedure. We have evaluated decision tree algorithm based on the following standard performance measures [33], [34]:

1) *True Positive (TP).* Number of the correctly predicted key fob
2) *True Negative (TN).* Number of the correctly predicted relay attack
3) *False Positive (FP).* Number of the incorrectly predicted key fob
4) *False Negative (FN).* Number of the incorrectly predicted relay attack
5) *True detection rate (TP rate).* Percentage of correctly predicted key fob, that is,

$$TPRate = \frac{TP}{TP + FN}. \tag{9}$$

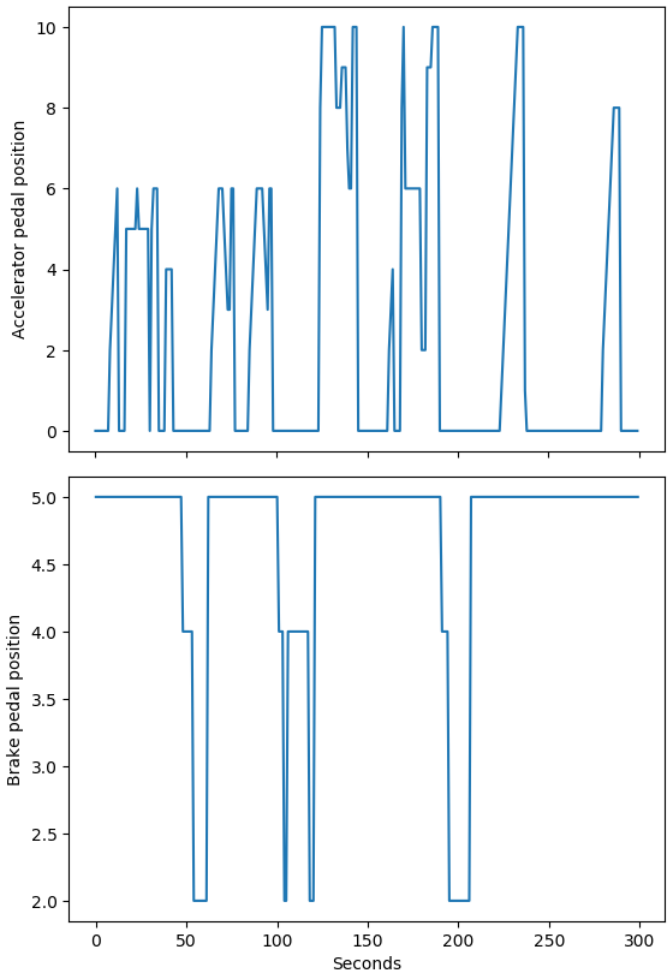

Fig. 5: An example of accelerator and brake pedal position of a 5-min (300 seconds) vehicle's trip

6) *False alarm rate (FP rate)*. Percentage of incorrectly predicted key fob, that is,

$$FPRate = \frac{FP}{FP + TN}. \tag{10}$$

7) *F-Measure*. Combines the recall and precision scores into a single measure of performance. F-measure is number between 0 and 1; the closer it is to 1, the better the accuracy of the test is and vice versa. F-measure is

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{11}$$

8) *Overall Accuracy*. Percentage of the correct prediction, that is,

$$OverallAccuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{12}$$

The 10-fold cross-validation eventually uses all observations in dataset for both training and testing. The accuracy of the decision tree is slightly improved to 99.8%, which was at 99% on train/test split. The accuracy rate of SVM and KNN has also improved and achieved 92.8% and 94.4% accuracy, respectively.

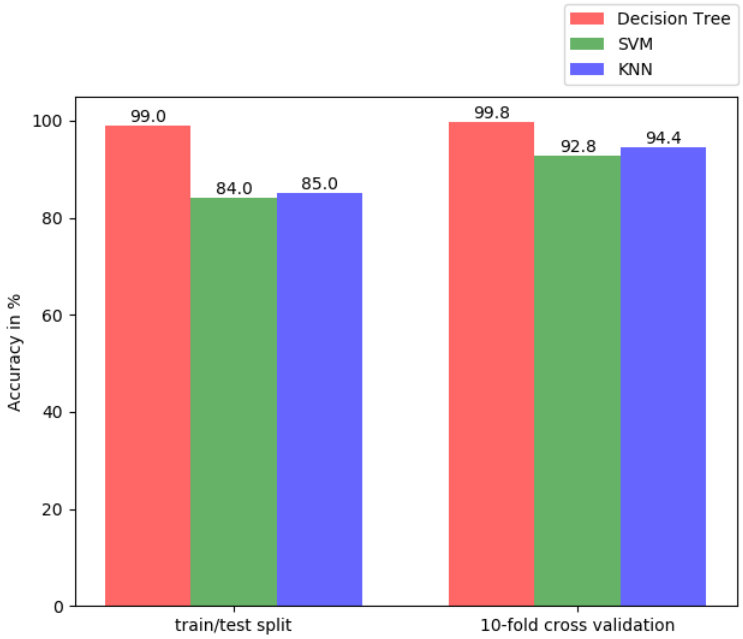

Fig. 6: Comparison of machine learning algorithm's relay attack detection accuracy rate

C. Neural Networks Results

In our recurrent neural network model, we have four hidden layers, including two LSTM layers and two dense layers. Both of the LSTM layers containing 64 hidden units (neurons), where first and second dense layers contain 32, and 1 hidden units, respectively. We used 400 iterations (epochs) to train our model. The feature vectors are sampled at regular 1 second time intervals with three features per vector, which are feeded into the recurrent neural network sequentially. The adam optimizer [35] is used with default arguments. Fig. 7 shows the accuracy of the LSTM recurrent neural network algorithm on each trip, and the overall achieved accuracy is $81\% \pm 7.5\%$.

V. Testing Environment

In this section, we outline the architecture of the PKES communication protocol, and our experimental setup using Arduino, that is an open source platform to test the scenario.

A. PKES Communication Protocol

The PKES system consists of a control module, a base station (vehicle), and a transponder (key fob). This system uses a low-frequency field to communicate with a key fob. The communication is based on a series of unidirectional data transfers between the key fob and vehicle. The control module is connected directly to the engine control module (ECU) and is used as an authentication unit to open the doors and also enable engine start. The control module issues challenges and evaluates responses from the key to enable or disable access. The vehicle acts as a gateway between the key fob and the control module. It communicates with the key via its physical interface. The key fob receives data from the vehicle, including commands and payload data and responds accordingly. The secret key that

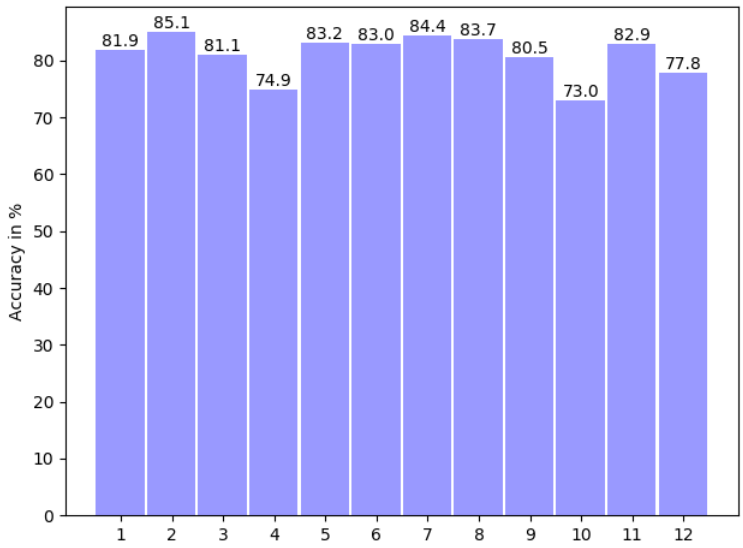

Fig. 7: LSTM recurrent neural network's driver identification accuracy on 12 trips

is internally stored in the key fob is used to encrypt challenges before the key fob replies with a response before authentication.

The PKES protocol varies among the car manufacturers. Also, different car models may use a different data structure for the PKES frame. Table III lists the size of PKES frame in different car models [36], [37]. Despite the slight variations in the protocol and data size, the overall structure of the PKES frame is almost the same, and it contains similar components. Fig. 8 depicts the general layout of PKES data, which is transmitted from the key fob to the vehicle through a unidirectional RF link. The protocol incorporates a secure rolling counter algorithm, where each message is unique in a sense that it includes an incrementing counter value. This can protect PKES communications from replay attacks [38], [39]. Each message contains a unique identification (UID), a counter value (CNTR), a command and control data (CMD), a message authentication code (MAC) which could be based on AES-128, and an error detection code such as a checksum (CHK). The PKES payload follows a preamble that is used for synchronization. The preamble precedes the transmission of the actual data which is used as a token for a receiver wake-up to detect an incoming data string.

Car manufacturers use various low power and high performance designs for PKES, some of which are ADF7023 [40] and Atmel ATA5795 [41]. Following the European ETSI EN300-220, the North American FCC (Part 15) and the Chinese short-range wireless regulatory standards [42], the key fob transceiver operates between 315 MHz to 928 MHz frequency bands, which is within the worldwide license-free industrial, scientific, and medical radio (ISM) bands. In addition, data (baud) rates may vary between 1 kbps to 300 kbps. Given that the maximum data rate of a PKES transceiver is 300 kbps and the minimum size of PKES frame is 60 bits (as shown in Table III), the transmission latency of PKES is at least 200 µs. In practice, the transmission latency could be higher, because the PKES signal should travel a distance between the Key fob and the antenna.

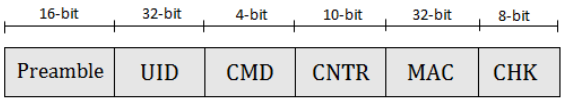

Fig. 8: PKES data structure

TABLE III: PKES frame size in different car models

Manufacturer	**Car model**	**Year**	**PKES frame size (bits)**
GAC Group	Trumpchi GS4	2017	104
BMW	X1	2014	80
Volkswagen	Golf Mk5	2004	60
Audi AG	TT	2008	96

B. Experimental Setup Using Real-World Data

We have used the Arduino, an open source platform to test the scenario. It supports both, tangible programmable microcontroller including the circuit board and a software programming language. Particularly, we have used two Arduino UNO boards, one to act as a key fob and second for the vehicle. We have used accelerometer and gyroscope sensor (GY-521 module) to detect key fob acceleration rate. The GY-521 module is an analog device which notes the corresponding values if the key fob moves in any direction. An accelerometer measures the non-gravitational acceleration when key fob moves from a standstill to any velocity, and gyroscope uses earth's gravity to determine the orientation. We also attached the GPS module with the key fob to determine the current location of the key fob. We have used the RF modules that are the small electronic devices to transmit and receive signals between the key fob and vehicle. RC Switch Arduino library is used to send and receive data over an RF medium. The RC Switch natively supports devices such as a 433 MHz AM transmitter and a receiver which we used for data transmission between the key fob and vehicle. The vehicle (RF receiver) periodically scans the key fob and transmits a wake-up call to determine the proximity of the key fob (RF transmitter). Once the key fob confirms its proximity, the vehicle sends a challenge message with its ID. Upon the receipt of the true response from the key fob, the vehicle unlocks itself.

1) Time and Space Analysis: Key fob acceleration and location are embedded (one bit for each) in the response message. The key fob transmits 1 or 0 as the key fob is accelerating or not, respectively. The vehicle sends its current location to the key fob while locking the vehicle when the user walks away or touches the car on exit. The key fob store this location and use it while sending the unlock signal. So at the time of unlocking, the key fob compares the current location of the key fob with the vehicles location. The key fob transmits 1 or 0 denote the key fob is near the vehicle or not, respectively. Despite the slight variations in the protocol and data size, the overall structure of the PKES frame is almost the same, and it contains similar components [36], [37]. We embed 1-bit for key fob acceleration (ACC) and 1-bit for location (LOC) in the PKES data frame, as depicted in Fig. 9.

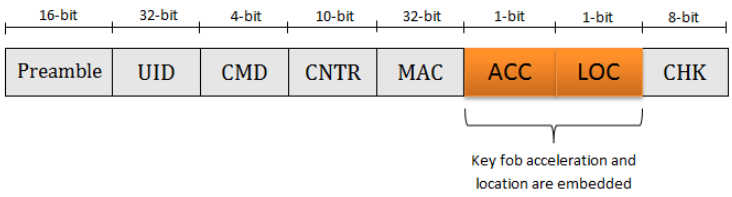

Fig. 9: PKES Test Frame

As discussed above, the decision tree has a fast prediction speed, that is, 1 ms for 100 test observations (we used 20% of dataset as a test dataset). Hence, it will take only 0.01 ms to predict one record.

2) Implementation Constraints: The implementation of our proposed solution is quick and effective. Location and key fob acceleration are implemented on the key fob side and transmitted to the car with the unlocking code. The rest of the five parameters and the machine learning algorithm are used inside the unlocking mechanism on the vehicle side. When the vehicle detects an irregular key fob behavior, it sends a passive response and will not unlock. The vehicle would also send a warning notification to owner's handheld device or other personal devices through the cellular network inside the vehicle. Although our proposed detection method has high accuracy, there might be false positive instances. In the case of false positives, we suggest the use of a button on the key fob to open the car, or the use of the owner's mobile phone to unlock the vehicle. To detect communication errors, we suggest the use of an error detection code, that is, checksum, in the key fob message packet.

VI. Conclusion

In this paper, we proposed a relay attack detection method by making use of a CART algorithm that uses seven security features for profiling normal key fob messages. The proposed algorithm can identify the legitimate drivers using three driving features and an LSTM recurrent neural network. We used a three-month log of a PKES system and the driving data is collected from a driver who drives the vehicles on 12 routes in real-world traffic conditions. First, a relay attack is detected using the CART algorithm to assess if the driving behaviors deviate from expected ones. We compared our CART algorithm with SVM and KNN learning algorithms, and the result of our tests demonstrated that our method outperforms other learning techniques. The proposed algorithm proceeds to gain the driving features for driver identification if a relay attack is not found. Our solution achieves the best relay attack detection accuracy rate, that is, 99.8%, and it can identify the legitimate drivers with 81% accuracy rate. One remaining challenge is to adapt the proposed solution for multiple drivers, as the proposed solution only uses a three-month data set of the key fob rather than the driver-specific data. As future work, we will to take into account the deep learning approaches to secure the remote attack vectors for modern vehicles such as Tire Pressure Monitoring System (TPMS), Bluetooth, Wi-Fi, Cellular, radio data system, and Telematics.

References

[1] "Internet crime complaint centre (IC3), motor vehicles increasingly vulnerable to remote exploits," *Available: http://www.ic3.gov/media/2016/160317.aspx*, Mar. 17, 2016.

[2] M. H. Eiza and Q. Ni, "Driving with sharks: Rethinking connected vehicles with vehicle cybersecurity," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 45–51, 2017.

[3] T. J. Waraksa, K. D. Fraley, R. E. Kiefer, D. G. Douglas, and L. H. Gilbert, "Passive keyless entry system," Jul. 17 1990, uS Patent 4,942,393.

[4] W. Choi, M. Seo, and D. H. Lee, "Sound-proximity: 2-factor authentication against relay attack on passive keyless entry and start system," *Journal of Advanced Transportation*, vol. 2018, 2018.

[5] K. Fu and W. Xu, "Risks of trusting the physics of sensors," *Communications of the ACM*, vol. 61, no. 2, pp. 20–23, 2018.

[6] M. Bacchus, A. Coronado, and M. A. Gutierrez, "The insights into car hacking," 2017.

[7] A. Jolfaei, A. Vizandan, and A. Mirghadri, "Image encryption using HC-128 and HC-256 stream ciphers," *International Journal of Electronic Security and Digital Forensics*, vol. 4, no. 1, pp. 19–42, 2012.

[8] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "A secure lightweight texture encryption scheme," in *Image and Video Technology*. Springer, 2015, pp. 344–356.

[9] A. Jolfaei and K. Kant, "Data security in multiparty edge computing environments," in *GOMACTech Conference, Artificial Intelligence Cyber Security: Challenges and Opportunities for the Government, Albuquerque, NM, USA*, 2019, pp. 17–22.

[10] A. Ranganathan and S. Capkun, "Are we really close? verifying proximity in wireless systems," *IEEE Security & Privacy*, 2017.

[11] R. Y. Asmar, D. T. Proefke, C. J. Bongiorno, and A. P. Creguer, "Method and system for authenticating vehicle equipped with passive keyless system," Jul. 18 2017, uS Patent 9,710,983.

[12] G. J. Udo, "Privacy and security concerns as major barriers for e-commerce: a survey study," *Information Management & Computer Security*, vol. 9, no. 4, pp. 165–174, 2001.

[13] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, 2015.

[14] A. Ur-Rehman, I. Gondal, J. Kamruzzuman, and A. Jolfaei, "Vulnerability modelling for hybrid it systems," in *2019 IEEE International Conference on Industrial Technology (ICIT)*, Feb 2019, pp. 1186–1191.

[15] A. Francillon, B. Danev, and S. Capkun, "Relay attacks on passive keyless entry and start systems in modern cars," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Eidgenössische Technische Hochschule Zürich, Department of Computer Science, 2011.

[16] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlidès, "Lock it and still lose it-on the (in) security of automotive remote keyless entry systems." in *USENIX Security Symposium*, 2016.

[17] L. Guan, J. Lin, B. Luo, J. Jing, and J. Wang, "Protecting private keys against memory disclosure attacks using hardware transactional memory," in *IEEE Symposium on Security and Privacy*, 2015, pp. 3–19.

[18] A. Jolfaei and A. Mirghadri, "Substitution-permutation based image cipher using chaotic henon and baker's maps," *International Review on Computers and Software*, vol. 6, no. 1, pp. 40–54, 2011.

[19] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "Canauth-a simple, backward compatible broadcast authentication protocol for can bus," in *ECRYPT Workshop on Lightweight Cryptography*, vol. 2011, 2011.

[20] B. Groza, S. Murvay, A. Van Herrewege, and I. Verbauwhede, "Libra-can: a lightweight broadcast authentication protocol for controller area networks," in *International Conference on Cryptology and Network Security*, 2012, pp. 185–200.

[21] O. Hartkopp and R. M. Schilling, "Message authenticated can," in *Escar Conference, Berlin, Germany*, 2012.

[22] A. Jolfaei and K. Kant, "Privacy and security of connected vehicles in intelligent transportation system," in *49th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2019)*, 2019.

[23] S. Ghane, A. Jolfaei, L. Kulik, and K. Ramamohanarao, "Differentially private streaming to untrusted edge servers in intelligent transportation system," in *18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2019.

[24] A. Jolfaei, K. Kant, and H. Shafei, "Secure data streaming to untrusted road side units in intelligent transportation system," in *18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2019.

[25] J. Park, Z. Chen, L. Kiliaris, M. L. Kuang, M. A. Masrur, A. M. Phillips, and Y. L. Murphey, "Intelligent vehicle power control based on machine learning of optimal control parameters and prediction of road type and traffic congestion," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 4741–4756, 2009.

[26] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 267–276, 2010.

[27] U. Ahmad, H. Song, A. Bilal, M. Alazab, and A. Jolfaei, *Secure Passive Keyless Entry and Start System Using Machine Learning: 11th International Conference and Satellite Workshops, SpaCCS 2018, Melbourne, NSW, Australia, December 11-13, 2018, Proceedings*, 12 2018, pp. 304–313.

[28] C. Saiprasert and S. Thajchayapong, "Remote driver identification using minimal sensory data," *IEEE Communications Letters*, vol. 19, no. 10, pp. 1706–1709, 2015.

[29] T. Tanprasert, C. Saiprasert, and S. Thajchayapong, "Combining unsupervised anomaly detection and neural networks for driver identification," *Journal of Advanced Transportation*, vol. 2017, 2017.

[30] M. Zhang, C. Chen, T. Wo, T. Xie, M. Z. A. Bhuiyan, and X. Lin, "Safedrive: online driving anomaly detection from large-scale vehicle data," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2087–2096, 2017.

[31] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[33] M. Kaur, G. Kaur, P. K. Sharma, A. Jolfaei, and D. Singh, "Binary cuckoo search metaheuristic-based supercomputing framework for human behavior analysis in smart home," *The Journal of Supercomputing*, pp. 1–24, 2019.

[34] R. Vinayakumar, M. Alazab, A. Jolfaei, K. Soman, and P. Poornachandran, "Ransomware triage using deep learning: twitter as a case study," in *International Conference on Cybersecurity and Cyberforensics Conference (CCC)*, 2019, pp. 67–73.

[35] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[36] R. Benadjila, M. Renard, J. Lopes-Esteves, and C. Kasmi, "One car, two frames: Attacks on hitag-2 remote keyless entry systems revisited," in *11th USENIX Workshop on Offensive Technologies WOOT 17)*, 2017.

[37] H.-L. Liu, S.-Y. Zhu, Z.-J. Lu, Z.-L. Liu *et al.*, "Practical contactless attacks on hitag2-based immobilizer and rke systems," *DEStech Transactions on Computer Science and Engineering*, no. CCNT, 2018.

[38] A. Jolfaei and K. Kant, "A lightweight integrity protection scheme for fast communications in smart grid," in *International Conference on Security and Cryptography*, 2017, pp. 31–42.

[39] A. Jolfaei and K. Kant, "A lightweight integrity protection scheme for low latency smart grid applications," *Computers & Security*, vol. 86, pp. 471–483, 2019.

[40] Analog Devices, "High performance, low power, ism band fsk/gfsk/ook/msk/gmsk transceiver ic," in *Data Sheet*, 2012, pp. 1–113.

[41] J. Goings, T. Prescott, M. Hahnen, and K. Militzer, "Design and security considerations for passive immobilizer systems," *Atmel publication*, 2010.

[42] G. Retz, H. Shanan, K. Mulvaney, S. O'Mahony, M. Chanca, P. Crowley, C. Billon, M. K. Khan, J. J. L. Orive, and P. Quinlan, "Radio transceivers for wireless personal area networks using ieee802. 15.4," *IEEE Communications Magazine*, vol. 47, no. 9, pp. 150–158, 2009.