Tampering Detection for Automotive Exhaust Aftertreatment Systems using Long Short-Term Memory Predictive Networks

1st Roland Bolboacă
'G. E. Palade' University of Medicine, Pharmacy, Science and Technology
Târgu Mureş, Romania
roland.bolboaca@umfst.ro

2nd Piroska Haller
'G. E. Palade' University of Medicine, Pharmacy, Science and Technology
Târgu Mureş, Romania
piroska.haller@umfst.ro

3rd Dimitris Kontses
Aristotle University of Thessaloniki
Thessaloniki, Greece
dkontses@auth.gr

4th Alexandros Papageorgiou-Koutoulas
Aristotle University of Thessaloniki
Thessaloniki, Greece
alexandpd@auth.gr

5th Stylianos Doulgeris
Aristotle University of Thessaloniki
Thessaloniki, Greece
destylia@auth.gr

6th Nikolaos Zingopis
Exothermia SA
Thessaloniki, Greece
Nikos.Zingopis@exothermia.com

7nd Zissis Samaras
Aristotle University of Thessaloniki
Thessaloniki, Greece
zisis@auth.gr

***Abstract*—The act of tampering can be defined as a single event ranging from actions such as resetting the reading of an odometer to more advanced and long-term actions such as manipulation of the vehicle's emission control systems. Tampering, however, requires certain interventions and changes to be made on the vehicle. Recently, the sophistication of certain vehicle sub-systems such as the emission control system, have also increased the sophistication of the tampering devices. Nowadays, tampering involves not only physical changes to certain automotive sub-systems, but also the manipulation of communication signals in order to hide the presence of tampering devices. This paper presents a detection method addressing tampering of the Automotive Exhaust Aftertreatment Systems. The proposed approach leverages Long Short-Term Memory predictive networks as detection models together with Cumulative Sum control charts. The proposed detectors were validated on datasets produced by a state-of-the-art aftertreatment simulation model of a heavy-duty vehicle. The datasets encompass diverse driving scenarios alongside known and unknown (e.g., possible future) tampering methods.**

***Index Terms*—Anomaly detection, Tampering detection, Automotive, Long Short-Term Memory, Cumulative Sum**

1. Introduction

Vehicle tampering can represent a single event such as resetting the reading of an odometer [12], [25], or it may consist in a set of long-term actions, such as changes in the vehicle emission control systems [13]. In all cases, however, tampering requires certain interventions and changes to be performed on the vehicle. Recently, the sophistication of certain vehicle sub-systems, such as the emission control system (e.g., up to Euro VI emission norms), have also increased the sophistication of tampering devices. As a result, nowadays, tampering involves not only physical changes to certain sub-systems, but also the manipulation of communication signals, in order to hide the presence of tampering devices [14], [15].

The scale of tampering has been also illustrated by several official reports. For example, the European Commission estimated that up to 50% of second-hand cars that are being traded across the borders within the European Union (EU) have their odometer manipulated [16]. Subsequently, a study of the Danish Ministry of Environment and Food [17] showed the massive scale of tampering, where up to 25% of high emitting heavy-duty vehicles in Denmark may be subject to tampering. Other reports showed that a large number of trucks across various EU countries have emissions much higher than their Euro norm, which suggests the presence of tampering (or the lack of maintenance) with the vehicle's emission control systems [18].

Unfortunately, such manipulations can not be detected by the vehicle's On-Board Diagnostic (OBD) system, which is meant to provide self-diagnostic capabilities, and vehicle status updates. Furthermore, the complexity of tampering detection increases when the tampering approach (e.g., sensor emulator) is capable of controlling and disabling certain functions, by injecting allowed, legitimate data and commands to control units. The key challenge of tampering detection is that tampered data (e.g., signal data) is benign when observed alone, but anomalous when observed in relationship with other observations.

To tackle this issue, we propose a detection methodology targeting unknown tampering detection at application level. By doing so, the methodology is hardware and protocol agnostic. Our approach leverages a set of detectors, each meant to predict a tampered signal, based on a group of input signals available in the vehicle's network, correlated with the predicted one. A Long Short-Term Memory (LSTM) neural network is considered as

a main prediction model for each detector. Furthermore, to monitor changes in the prediction error of the LSTM network, a Cumulative Sum (CUSUM) is considered. The contributions of our approach are three-fold. First, in terms of detection, we focus on unknown tampering. Here, by only learning benign data, our detectors are able to identify signal tampering in time series with negligible delay since the tampering starting point, as demonstrated in the experiments. Second, to achieve this, our detectors take into consideration multiple past observation for each input signal, to predict the next signal value using LSTM networks, and then detect tampering based on a CUSUM approach. And last, our approach was validated on data produced by a state-of-the-art aftertreatment simulation model of a heavy-duty vehicle, thus it's viability in terms of real-world tampering detection scenarios stands out.

For the experimental assessment, both the benign, clean measurements, together with the tampered ones were produced by a state-of-the-art aftertreatment simulation model of a heavy-duty vehicle. Our experiments show fast detection times for each tampered measurement, with delays in the order of seconds for the starting point of the tampering to the detection moment.

The rest of the paper is organized as follows. Section 2 provides a background on Exhaust Aftertreatment Systems. Section 3 outlines a series of relevant related studies for the paper at hand. This is followed by Section 4, where the detection methodology is presented, then, afterwards, in Section 5, a detailed description of: the simulator, the used dataset and the experimental results is provided. The paper concludes in Section 6.

2. The Exhaust Aftertreatment System

The main purpose of the Exhaust Aftertreatment System (EAS) is to reduce the amount of pollutants generated by diesel vehicles. To achieve this, early EAS used metal catalysts to oxidize carbon monoxide, and hydrocarbons, to reduce them into carbon dioxide and water. From there on, the EAS was continuously improved with additional catalytic converters, analog and digital sensors, and computer controlled processes. Consequently, this continuous improvement of the EAS gradually reduced the pollutants discharged by diesel vehicles, and decreased the equipment production costs [11].

Today, the primary objective of the EAS remains the reduction of pollutants generated by the engine to less harmful elements. These pollutants include: carbon monoxide (CO), hydrocarbons (HC), nitrogen oxides (NOx) and particulate matter (PM). While passing through the EAS, the contaminants are converted into carbon dioxide (CO_2), water (H_20) and nitrogen (N_2). These reduction operations are carried out by the catalytic converters inside the EAS. A catalytic converter is an umbrella term encompassing multiple emission control devices which use a catalyst to facilitate a chemical conversion of pollutants into less harmful elements. Some of the catalytic converters are: Oxidation catalysts (oxycats), Three-way catalysts (TWCs), Lean NOx traps (LNTs), Particulate Filters (PF), Selective Catalytic Reduction (SCR) systems or Ammonia Slip Catalysts (ASCs) [11]. A general overview of a diesel EAS that contains the most common elements, is shown in Figure 1.

While the technological advancements of the EAS, together with the ever more strict regulations, considerably reduced the emission levels, illegal manipulations of the EAS are still found during vehicle inspections. As described in [14], multiple EAS components, such as SCR systems, Diesel Particle Filter (DPF) and Exhaust Gas Recirculation (EGR) systems, are tampered with hardware manipulations (e.g., emulators, modifiers, OBD Suppressors). Most emulators used on heavy-duty vehicles target the SCR system or the NOx sensors. The main intention here is to stop the dosing of the Diesel Exhaust Fluid (e.g., AdBlue) required by the SCR to reduce NOx emissions.

The problem becomes more severe when fleets of vehicles have their EAS tampered over long periods of time. This infringement of environmental regulations, in most cases, doesn't require expert knowledge. Simple emulators can be purchased at low prices from public e-commerce web sites, and then, they can be easily installed by simply connecting them to the OBD port of the vehicle. More complex emulators can be attached directly to the vehicle's internal network/bus and, in some cases, may require additional manipulation of other hardware or software components. Furthermore, the large number of tampering techniques and devices available on the market is continuously increasing, and in the end, it becomes difficult for authorities to keep up with innovating tampering approaches.

3. Related Work

In the scientific literature only a handful of papers address the problem of tampering detection. Tampering detection is, however, associated to anomaly detection, considering only the cases that do not cause parameter deviations outside the normal functioning ranges. This section provides an overview of existing tampering detection techniques, together with anomaly detection methods targeting automotive systems.

Recently, Roman *et al.* [28] proposed a privacy-preserving tampering detection method for light duty vehicles. Here, the authors proposed a Fast Fourier transform based distortion technique which preserves the privacy of sensor data collected from a vehicle. Moreover, the authors deployed a Random Forest regression method able to detect tampering on both clean and anonymized data. While their proposed solution showed promising results in terms of detection detection, the main contribution of the paper remains the data distortion procedure to preserve data privacy. Subsequently, the validation was performed with synthetically generated tampering by replaying previously recorded values. Conversely, the tampering scenarios considered in the work at hand include a large palette of manipulations that could be performed by future tamperers.

Similarly, Bolboacă *et al.* [29] proposed a Locality Sensitive Hashing (LSH) based approached for detecting tampering in multi-dimensional data. Their approach is based on the proprieties of LSH, namely, the high probability that two points close to each other in Euclidean space hash to similar values. By leveraging this property, the detection method demonstrated that even small deviations in tampered data, produce high deviations in the LSH

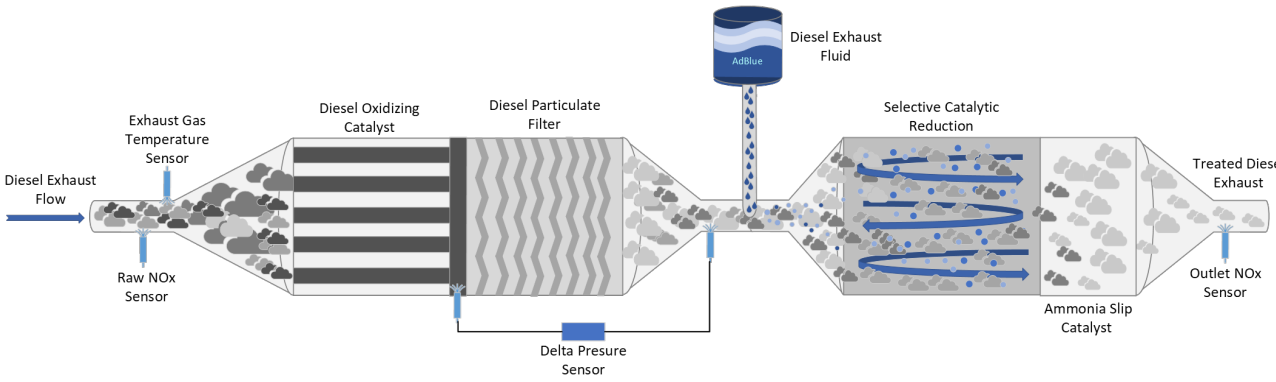

Figure 1. General overview of a diesel exhaust aftertreatment system.

method, thus the probability of collision of the tampered data-points would decrease significantly.

Moving towards anomaly detection techniques, we find the work of Groza and Murvay [31]. Here, a replay and modification detection technique, based on the number of Control Area Network (CAN) frames sent, their periodicity and the data-field entropy was proposed. The approach leverages the Hamming distance between two frames, originating from the same sender. As such, this anomaly detection was shown to be capable of detecting random changes in the frame's content. Likewise [31], Stabili *et al.* [30] proposed an anomaly detection algorithm for the CAN bus based on the Hamming distance between the payloads of two consecutive CAN messages with the same identifiers. Their proposed approach is focused on detecting fuzzy, as well as replay attacks.

In terms of more advanced detection algorithms, we find the work of Tianjia He *et al.* [34] where an autoencoder Neural Network was used to detect anomalies on multiple features. Next, Longari *et al.* [32] proposed an Intrusion Detection System (IDS) solution for the CAN bus, where LSTM-autoencoders were considered to detect abnormal traffic patterns. In the same direction, we find the work of Shin *et al.* [27], where an LSTM-based sensor attack detection method was considered as a response to deception attacks in anomalous vehicles.

Switching our focus to predictive models for EAS, we find various solutions based on neural networks [19]–[21], [23], [24]. All of them leverage the capabilities of neural networks for the prediction of emissions such as oxides of nitrogen (NOx) and particulate matter (PM), on different types of automotive engines. In a similar way, neural networks have been successfully proposed for other prediction tasks, such as vehicle speed prediction using LSTM networks [26] and air-fuel ratio (AFR) prediction in spark-ignited engines using Recurrent Neural Networks [22]. se studies applied LSTM as tampering detection technique.

In contrast to previous works and existing solutions, this paper brings several contributions. To begin with, we believe this might be one of the first approaches explicitly addressing the detection of tampering in EAS. Second, while several studies used real data for model training, most of the attacks and the tampering have been synthetically generated. In contrast, in the paper at hand, the dataset used in the experimental assessment comprises a state-of-the-art recreation of the aftertreatment system of a Renault MDA2C EuroV HD Vehicle. Lastly, while the proposed approach focuses on tampering detection in the context of the EAS, the approach may find applications for tampering detection in other vehicle subsystems as well (e.g., odometer tampering).

4. Proposed Approach

In brief, the proposed detection method deploys a set of detectors, where each detector encompasses a Long Short-Term Memory (LSTM) predictive network, accompanied by a group of input signals correlated to an output signal, a Cumulative Sum (CUSUM) chart and its detection threshold used to determine if a presented observation is tampered or not. Consequently, each deployed detector monitors a single signal (e.g., Outlet NOx, Urea dosage). Considering LSTM networks, the envisioned detectors present a high degree of flexibility in terms of the chosen prediction method. Each detector can operate differently, such as: one-to-one, many-to-one, many-to-many, sliding window one-to-one, sliding window many-to-one and sliding window many-to-many, as showcased in the following subsections.

The proposed approach is envisioned to function as in-vehicle detectors positioned at application level, namely as Electronic Control Units (ECU) applications, without requiring direct access to the CAN bus. Distancing the approach from the CAN bus level brings several advantages, such as having the ability to function on top of different communication protocols (e.g., LIN, CAN, FlexRay, Ethernet) without knowledge of CAN frame structure or frequency.

For the most part, in terms of computational resources, the capabilities of the ECUs are extremely limited. Furthermore, the training procedure of the LSTM networks can be a slow and high resource demanding process. Considering these limitations we propose an offline training methodology. Namely, the LSTM networks are trained outside of the vehicle with clean data provided by measurements. The trained models (i.e., the neural network's weight matrices) are later deployed onto the vehicle's ECU.

4.1. Signal Analysis

A crucial part in the proposed method consists of selecting the appropriate signals for the LSTM networks, so that the used input signals are correlated to the output signal that presents interest in detecting tampering (e.g., NOx signal). The selection process leverages Pearson's product-moment correlation coefficient [2], in order to select from a large number of measured signals, only those signals that exhibit the highest correlation coefficient associated with the chosen output signal. Pearson's product momentum correlation (Pearson's correlation) is a statistical measure describing the strength of the linear relationship between two variables. In short, the change in the magnitude of one variable can be related with a change in the magnitude of another variable.

As a first step in the selection process, the output variable (i.e. the monitored signal) for each detector is chosen. For each selected output signal, the corresponding group of input signals is chosen based on the highest positive, as well as, negative correlation coefficients.

Consider X of length n, the set containing all the available signals (variables) measured from a vehicle, where n denotes the total number of measured variables. We define Y as a subset of X, such that $Y \subseteq X$, contains the selected output variables. Furthermore, we consider m as the total number of LSTM networks used in the detection process. For each network j, consider the output $y^j \in Y$ that can be selected as input for the other networks as well. By considering K the number of samples/measurements taken for each variable, given y^j and $x^i \in X$, where $i = 1..n$, the correlation coefficient $R(y^j, x^i)$ is computed as:

$$R(y^j, x^i) = \frac{\sum_{l=1}^{K}(y_l^j - \overline{y^j})(x_l^i - \overline{x^i})}{[\sum_{l=1}^{k}(y_l^j - \overline{y^j})^2 \sum_{l=1}^{K}(x_l^i - \overline{x^i})^2]^{1/2}}, \quad (1)$$

where $\overline{x^i}$ and $\overline{y^j}$ represent the mean values of variable x^i and y^j respectively. The possible values for R are between $[-1, 1]$ and the sign of R signifies either a positive or negative correlation between the two variables, that is, a change either in the same or in the opposite direction. In the following, X^j, where $j = 1..m$, will denote the set of selected inputs for network j.

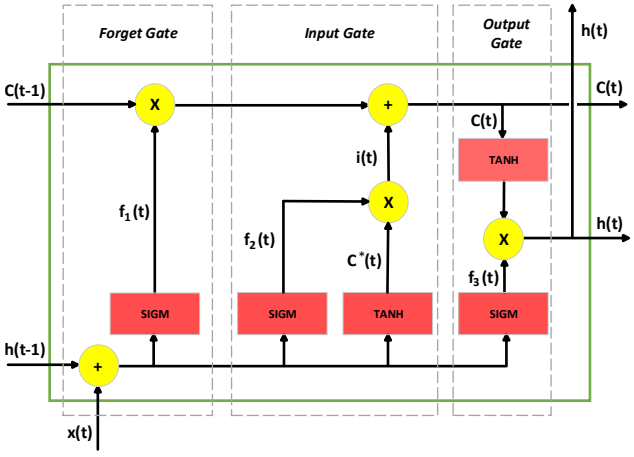

Figure 2. Long Short-Term Memory network block architecture.

4.2. Detector Design

The fundamental components of an LSTM Network are a sequential input layer and an LSTM layer. The scope of the sequential layer is to feed data sequences or time series to the network, while the LSTM layer is responsible for learning the long-term dependencies between the time steps of the sequence input data. Though this is the basic architecture of an LSTM network, the concept of deep learning can also be applied for this type of neural network by adding multiple LSTM layers.

The typical LSTM layer incorporates blocks. As depicted in Figure 2, they are mainly composed of one or more memory cells, an input gate, an output gate and a forget gate. The cell *remembers* the information over time while the gates regulate the information flow coming in and out of the memory cell.

The forget gate controls the information which is to be discarded from the cell state. For a given unit t, the forget gate is denoted as $f(t)$, and is expressed as:

$$f_1(t) = \sigma(w_{f_1}(x(t) + h(t-1)) + b_{f_1}), \quad (2)$$

where w_{f_1} denotes the weight matrix for $f_1(t)$, $x(t)$ represents the input vector, b_{f_1} denotes the bias vectors and $h(t-1)$ is the previous hidden state of the memory cell. The result is *squashed* in the range $[0, 1]$ using the sigmoid activation function, denoted as σ in the following equation:

$$\sigma(x) = \frac{1}{1 + e^x}. \quad (3)$$

The input gate $i(t)$, controls the new information which is to be saved. It is composed of a second forget gate (having a separate weight matrix), responsible for regulating how much information from the input will be stored in the cell state, and a module responsible for creating the new candidates for the cell state. The input gate equations are as follows:

$$i(t) = f_2(t) \cdot \overline{C}(t), \quad (4)$$

where $f_2(t)$ denotes the second forget gate defined as:

$$f_2(t) = \sigma(w_{f_2}(x(t) + h(t-1)) + b_{f_2}). \quad (5)$$

Following, in equation 4 the candidates are denoted as $\overline{C}$ and are computed as follows:

$$\overline{C}(t) = tanh(w_C(x(t) + h(t-1)) + b_{\overline{C}}), \quad (6)$$

here, $tanh$ denotes the hyperbolic tangent (tanh) activation function, where tanh is defined as:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (7)$$

The new cell state, denoted as $C(t)$, is computed as follows:

$$C(t) = f(t) \cdot C(t-1) + i(t), \quad (8)$$

where, $C(t-1)$ denotes the previous cell state.

The output gate computes the unit's output based on the current cell state. Here, a third forget gate is used, denoted as $f_3(t)$. The equation for the output gate is as follows:

$$f_3(t) = \sigma(w_{f_3}(x(t) + h(t-1)) + b_{f_3}), \quad (9)$$

where:

$$h(t) = f_3(t) \cdot tanh(C(t)). \quad (10)$$

In Equation 10 the unit's output, which is also a hidden state, is denoted as $h(t)$, while the current cell state is denoted as $C(t)$.

4.3. CUSUM and Threshold Computation

For monitoring the changes in the LSTM network prediction error a CUSUM based approach was selected, namely the 1-CUSUM scheme [1]. Compared to the typical CUSUM control charts the 1-CUSUM scheme is able to detect the changes (i.e., increase and decrease shift) in both the mean as well as the variance values, using only one two-sided CUSUM control chart.

Let μ_0 denote the mean value of the prediction error, computed during the training phase. Let e denote the prediction error computed during the detection phase. Also, let $v = e - \mu_0$. The CUSUM control chart for detector j, denoted as B_i^j, at time i, is defined as:

$$\begin{aligned} &B_i^j = max[0, B_{i-1}^j + (\lambda v_i + (1-\lambda)v_i^2) - \beta_B] \\ &\texttt{if } (B_{i-1}^j < 0) \texttt{ or } (B_{i-1}^j = 0 \texttt{ and } v_i > 0) \\ &\texttt{or} \\ &B_i^j = min[0, B_{i-1}^j + (\lambda v_i - (1-\lambda)v_i^2) + \beta_B] \\ &\texttt{if } (B_{i-1}^j < 0) \texttt{ or } (B_{i-1}^j = 0 \texttt{ and } v_i < 0). \end{aligned} \quad (11)$$

In Equation 11, β_B is the reference parameter value, λ ($0 \leq \lambda \leq 1$) is the weighting factor and B_0^j is initialized with 0.

The detection threshold for each detector, denoted as θ_j, is computed after training each network. Firstly, each network performs a forward pass on the previously trained data. Secondly, for each new prediction, y_i^j, the prediction error is computed. This step is followed by the computation of B_i^j on all the training prediction errors. Finally, the value for θ_j is computed as:

$$\theta_j = max(B_i^j), \quad (12)$$

4.4. Tampering Detection

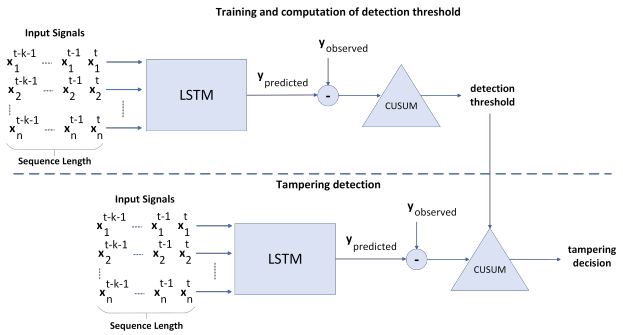

Figure 3. Overview of the proposed approach.

The complexity of tampering detection spawns sophisticated methods that must be applied in order to obtain significant results. With this in mind, a multi-phase tampering detection approach is formulated.

Let $d_j \in D$ denote a detector, where D represents the set of all the detectors. Each d_j incorporates the following components: a group of input signals X^j, an LSTM network, a predicted output signal y^j, a CUSUM B_j, and a detection threshold θ_j. A general overview of the proposed approach is shown in Figure 3.

During the detection phase, for each detector d_j, the CUSUM value B_0 is initialed to 0. Once new observations are available, each detector's LSTM network performs a forward pass using the newly received data. After each pass, the prediction error and the new B_j values are computed. Here, the prediction error is expressed as the difference between the actual value (i.e., the real value or the emulated value in case of tampering) and the predicted value y_i^j. Each detector generates an alert once the value of B_j exceeds the threshold θ_j.

As already mentioned, each detector can operate in several ways (e.g, one-to-one, many-to-one, sliding window one-to-one). This features refers to the way the data is fed into the neural network and not to the number of inputs and outputs it has, namely it refers to networks that can work with sequential, temporal data.

Recall, the fundamental components of each LSTM network are the sequential input layer and the LSTM layer. In this respect, the sequential input layers are capable of feeding entire data sequences to the LSTM layer. When one-to-one method is used, the input data will contain only one time step t and the network will predict a single value for the next time-step $t+1$. When many-to-one method is used, the input data will contain a sequence of multiple time-steps $[t-k, t]$ and the network will predict a single value for the next time step $t+1$. Lastly, the many-to-many method involves feeding a sequence containing k time-steps while the network outputs another sequence also containing k time-steps.

In a real-time detection scenario, for the many-to-many and many-to-one methods, the computation of B_i^j would not be performed for each new observation but rather after k new observations are available. Here, one must also consider that the on-board sensing devices (e.g., sensors) send information at different rates. This would naturally introduce a certain delay, first, in the computation of the CUSUM value and then, in the detection procedure.

Moreover, the length of the input sequence, previously denoted as k, directly influences the training/validation errors (e.g., RMSE), the prediction errors and consequently, the detection process. Therefore, choosing the best value for k is imperative for the performance of the proposed approach. The influence of k on both the training/validation RMSE as well as on the detection threshold θ is shown in Figure 4. Here, the same LSTM neural network is trained, using the same training dataset and the same hyper-parameters, including the same initial weights, however, different values for k are used.

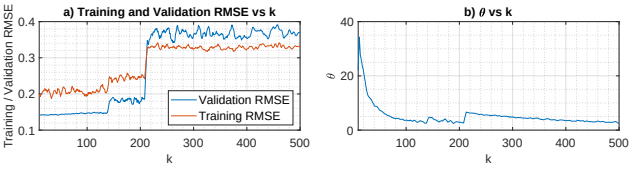

Figure 4. Illustration of how the length of the input sequence influences the training/validation RMSE and the detection threshold.

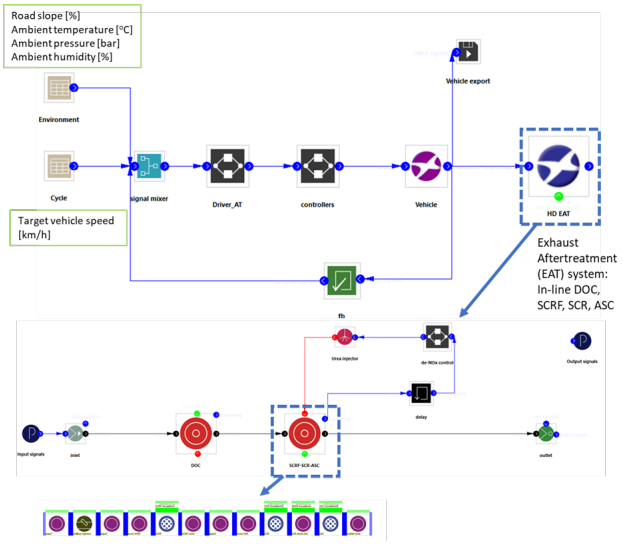

Figure 5. The simulation model for HD vehicle (Bus) using Exothermia suite.

5. Proposed Detectors Evaluation

To expand the evaluation of the anomaly detection algorithm/framework and test its applicability, a simulation approach was followed. The goal was to produce the necessary data derived from a vehicle layout different from the one used for training the model. To that aim, a simulation model of a heavy-duty vehicle was developed in the Exothermia suite simulation environment [33]. Exothermia suite is a simulation platform that includes solvers for physicochemical modelling of flow systems of particular relevance in emission control systems. Exothermia suite can also function as a co-simulation host for 3rd party models conforming to the Functional Mock-up Interface (FMI) standards.

Furthermore, a prototype of the proposed EAS tampering detection method was implemented in Matlab R2021a. This section describes the dataset used for validation, the tampering scenarios, the architecture of the detectors and finally the results of the evaluation.

5.1. Dataset and Simulation Model Description

The simulation model, as presented in Figure 5, represents a bus that incorporates a state-of-the-art exhaust aftertreatment system with in-line Diesel Oxidation Catalyst (DOC), Selective Catalytic Reduction Filter (SCRF), Selective Catalytic Reduction system (SCR) and Ammonia Slip Catalyst (ASC). For the deNOx control, a closed-loop coverage-based Ammonia (NH_3) control strategy is

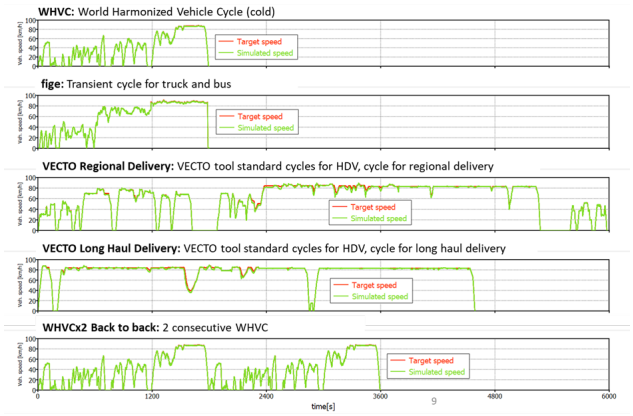

Figure 6. Simulated driving cycles on the Bus model.

followed. Thus, deNOx performance is limited by NH_3 slip, which must be retained below a certain limit (e.g. 10 ppm). The model was validated with experimental data. The main inputs of the model, regarding the mission profile, are the target vehicle speed, the road slope, and its environmental conditions (ambient temperature, pressure, and humidity). The output of the model includes, among others, the exhaust gas composition and aftertreatment parameters (gaseous emissions, exhaust gas temperature, urea dosing, etc). Gaseous emissions are calculated based on the emission profile's inputs, the integrated engine maps and the physicochemical solvers of the exhaust aftertreatment system components.

To provide results representing a variety of driving conditions, five different cycles were simulated:

The World Harmonized Vehicle Cycle (WHVC) is a chassis dynamometer test for heavy-duty vehicles occasionally used to compare the respective vehicle and engine emission levels for research purposes. It is developed based on the World Harmonized Transient Cycle (WHTC) which is used for engine emission certification/type approvals worldwide [35]. The duration of the cycle is 1800 seconds including three segments, representing urban, rural and motorway driving. The WHVCx2 Back to back is cycle comprised of two consecutive WHVCs.

The Fige transient cycle is a transient test cycle for truck and bus engines developed by the former FIGE Institute, Aachen, Germany, based on real road cycle measurements of heavy-duty vehicles [36]. It is a non-standardized vehicle version of the European Transient Cycle (ETC) which is used for emission certification of heavy-duty diesel engines in Europe starting in the year 2000 [37]. Similar to WHVC cycle, Fige cycle is 1800 seconds long including urban, rural and motorway driving parts.

For the two VECTO standard cycles two VECTO mission profiles were simulated representing Long Haul and Regional delivery driving. Vehicle Energy Consumption calculation TOol (VECTO) was developed by the European Commission for determining CO_2 emissions and fuel consumption from Heavy-Duty Vehicles and was introduced in [38]. VECTO simulates CO_2 emissions and fuel consumption based on vehicle longitudinal dynamics using a driver model for backward simulation of target

speed cycles.

The simulated vehicle speed is calculated considering the target vehicle speed, kinematic equations, aerodynamic resistance, rolling friction resistance and losses of the drive system. From Figure 6 it can be observed that, in all cycles, the simulated speed perfectly fits the target one.

5.2. Tampering Scenarios

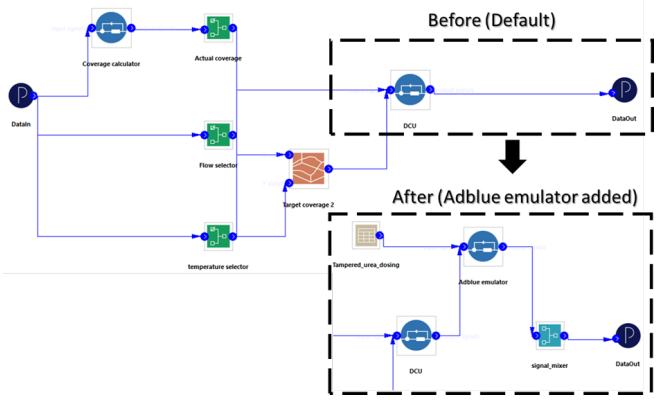

Figure 7. Simulation of the AdBlue emulator: Modifications in the simulation environment.

As a first step, all cycles were simulated without any modifications to the model to represent the non-tampered (baseline) scenarios. These baseline results were provided to train the detection model. The next step was to form tampering scenarios including known (observed) and unknown (possible future) tampering methods. All tampering scenarios and approaches consist of two main actions: the disabling of the targeted component and the hiding of the affected signals by providing emulated signals instead of the real ones. A future advanced emulator could occasionally work in a training mode where it tries to fit the emulated signals to the real ones. A well-known and frequently observed tampering attack in heavy-duty vehicles is the disabling of the deNOx system to obtain a decrease of operational costs needed like AdBlue consumption and refilling. Such a case was simulated in the bus model. To mitigate the AdBlue consumption, an AdBlue emulator, previously tested on a Renault MDA2C EuroVI truck, was simulated while both already observed and possible future methods were applied to hide the affected signals.

The main disabling action of the AdBlue emulator is that it reduces the urea dosing command while it creates two hiding signals regarding feedback to the Engine control module. Thus, in the simulation environment, the new signals simulated was the reduced urea dose and the hidden signals to the Engine control module. For this purpose, a new component representing the AdBlue emulator was added to the deNOx control system of the model as can be seen in Figure 7.

The emulation ("hiding") methods applied include known, simple in approach, methods like NOx downstream of SCR emulation as a percentage of the upstream sensor, and possible future more advanced methods like providing the emulated signals via multiple linear regression and moving average. Indicative results can be seen in Figure 8. Here the Outlet NOx signal without tampering and with three hiding methods is showcased.

Such advanced models aim to give emulated signals' patterns closer to the actual ones measured for a "working-ok" exhaust aftertreatment system. In this way, they increase the possibility to stay undetected from plausibility diagnostics and other tampering detectors. However, they also require more computational resources and additional signals should be available to train them efficiently. Furthermore, all "hiding" algorithms (known and possible future) were active when the actual urea dosing command was non-zero. The simulations were performed using 10Hz sampling frequencies for the used signals.

5.3. Detectors Architecture

For the experimental assessment two detectors were chosen. The first detector, denoted as d_1, monitors the Outlet NOx and the second detector, d_2, monitors the Urea dosage command. The architecture of each detector's LSTM network includes a sequence input layer containing five neurons for d_1 and eight neurons for d_2. It's worth mentioning that the number of neurons on the input layer were automatically determined during the signal analysis, based on the number of X^j input signals. For both detectors a prediction method of many-to-one has been selected, with an input sequence length k, containing 128 time steps.

Beside this, the rest of the LSTM network configuration is identical for both detectors. Namely, the second layer, consists of 16 LSTM cells which are fully connected to the input layer. Additionaly, the LSTM layer also contains a bias vector which was initialized to zero. The third layer is a fully connected layer consisting in 1 neuron, corresponding to the predicted output signal y^j. Furthermore, the output neuron is connected to all previous LSTM cells. Lastly, in the final layer, a regression layer is considered which uses the Root Mean Squared Error (RMSE) loss function for regression tasks. For the LSTM layer the default activation functions were used, namely: the Hyperbolic Tanget Function for the state activation and the Sigmoid function for gate activation.

Following an exhaustive methodology, the best results for the hyper-parameters, while using the adam optimizer [3], include 500 training epochs with an initial learning rate of 0.015. The learning rate is periodically dropped by 20% every 100 epochs. Moreover, to avoid over-fitting, the training procedure includes validation steps at a frequency of 10 epochs. If the validation loss value increases, the training process is stopped.

The outputs, y^j, and the corresponding input groups X^j were chosen for both detectors according to the selection methodology showcased in section 4.1. The selected input/output signals have been summarised in Table 1

5.4. Results

The current section presents the obtained results for the proposed tampering detection method. Some of the signal names as well as detailed description of the tampering scenarios have been purposefully concealed in order to ensure the paper at hand would not encourage future malevolent agendas. The experiments were conducted on

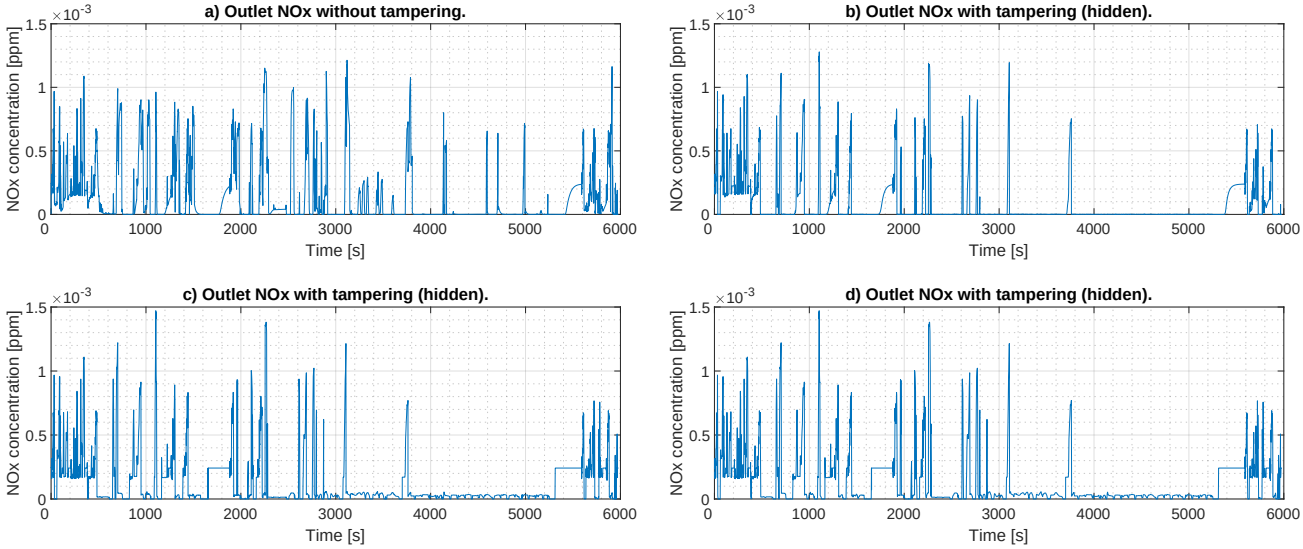

Figure 8. Illustration of the Outlet NOx signal in a baseline scenario together with three different hiding techniques.

TABLE 1. SELECTED INPUTS/OUTPUTS FOR BOTH DETECTORS

Detector	Output y^j	Input group X^j
d_1	Outlet NOx [ppm]	Engine Torque [Nm] SCR Temp. [°C] SCR Inlet NOx [ppm] Urea dosage command [%] SCR Twall Temp. [°C]
d_2	Urea dosage command [%]	Vehicle Speed [km/h] Accelerator Pedal [%] DOC Inlet Temp. [°C] SCRF Inlet Temp. [°C] SCR Inlet Temp. [°C] ASC Inlet Oxygen [%] DOC Pressure [Pa] SCRF Pressure [Pa]

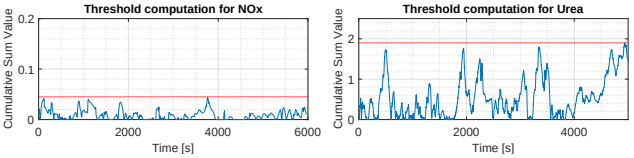

Figure 9. Threshold computation for Outlet NOx and Urea dosage command.

the datasets obtained from the simulation environment, as documented in Section 5.1. The simulation environment runs a state-of-the-art model recreating the EAS of a Renault MDA2C EuroV HD vehicle as well as the existing and possible future tampering methods.

Overall, a total of 75 experiments have been performed on datasets simulating five vehicle driving cycles and integrating several distinct hiding methods for both the NOx emissions and the Urea dossing command.

Both detectors were trained using a non-tampered scenario, namely the VECTO Regional Delivery cycle. The computed thresholds for both detectors, d_1 monitoring the Outlet NOx signal and d_2 monitoring the Urea Dosage Command signal, are showcased in Figure 9.

The results for the first experiment, for detector d_1,

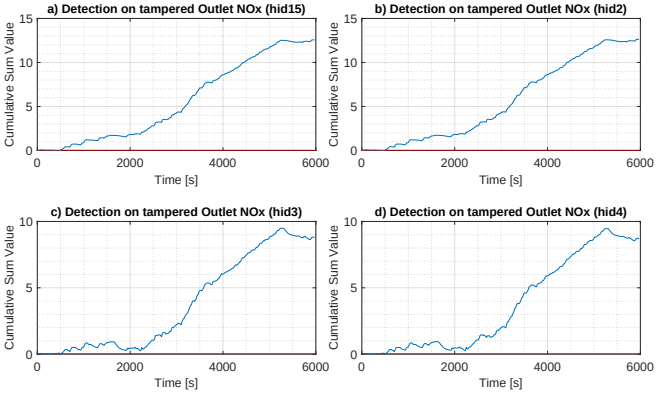

Figure 10. Illustration of the outcome of the tampering detection when the NOx signal is used as the predicted parameter. The four sub-figures illustrate detection in the case of four different NOx signal hiding techniques.

where the monitored signal is the Outlet NOx are shown in Figure 10. Here, four tampering hiding methods are presented for the VECTO Regional Delivery driving cycle. In this scenario, only the output (predicted value) was tampered. The four sub-graphs are illustrating the computed CUSUM for each of the tampering scenario. It can be observed that in this scenario the computed detection CUSUM exceeds the threshold early in the experiment in all four cases.

Concurrently, the results of another experiment for the same detector d_1, considering the same driving cycle, VECTO Regional Delivery, are showcased in Figure 11. Here, the output (predicted) signal is hidden, furthermore, one of the inputs is tampered. As in the previous experiment, the detection CUSUM exceeds the threshold early on in the experiment in all four tampering cases.

Advancing to the evaluation of the proposed approach for detector d_2, which monitors the Urea dosage command, for the VECTO Long Haul scenario. The detection results of four hiding methods is shown in Figure 12. Here,

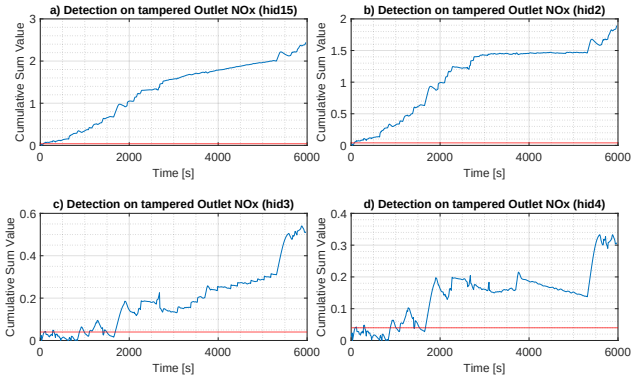

Figure 11. Illustration of the outcome of the tampering detection when the NOx signal is used as the predicted parameter and one of the input signals is tampered. The four sub-figures illustrate detection in the case of four different NOx signal hiding techniques.

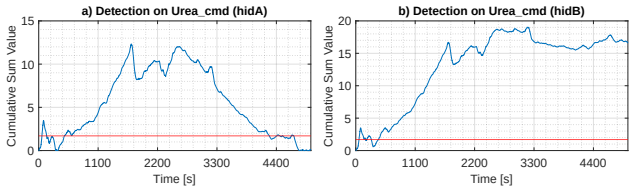

Figure 12. Illustration of the outcome of the tampering detection when the Urea dosage command is used as the predicted parameter. The two sub-figures illustrate detection in the case of two different Urea dosage command hiding techniques.

yet again, the two sub-graphs show distinct scenarios for hiding the Urea dosage command signal. In this scenario only the predicted signal (e.g., Urea dosage command) was tampered.

Finally, the results for all 75 experiments, in terms of minimum, maximum and average detection delay, are detailed in Table 2 and Table 3. The results for the first major scenario where one of the input signals is tampered and the output (predicted) value is hidden are detailed in Table 2. The results for the second major scenario where only the output (predicted) value is hidden while the inputs are untampered, is detailed in Table 3. In both tables the proposed detection method is evaluated on all five driving cycles, namely, VECTO Long haul, VECTO Regional delivery, WHVC, WHVCx2 Back to back, and Fige.

6. Conclusions

This paper approached a new and emerging threat for automotive systems, a threat leading to excess emissions of nitrogen oxides (NOx), particulate matter (PM), and other pollutants into the atmosphere. This threat has been identified as tampering. Furthermore, this paper presented a in-vehicle tampering detection methodology for heavy-duty vehicle Exhaust Aftertreatment Systems which is based on LSTM neural networks. While the proposed approach is focused on the EAS it's easily extendable to other automotive sub-systems as well. The applicability and effectiveness of the proposed approach has been demonstrated on datasets produced by a state-of-the-art

TABLE 2. SUMMARY OF TAMPERING SCENARIOS, DRIVING CYCLES, AND DETECTION TIME. HIDDEN SIGNAL: NOX (A TOTAL OF 60 EXPERIMENTS, WITH DIFFERENT HIDING TECHNIQUES).

Tampering scenario	**Driving Scenario**	**Detection Delay [s]**		
		Min.	**Max.**	**Avg.**
Tampered input and Hidden NOx signal	VECTO Long haul	72	72	72
	VECTO Regional delivery	63	81	75
	WHVC	50	60	51.66
	WHVCx2 Back to back	45	45	45
	Fige	30	30	30
Hidden NOx signal	VECTO Long haul	99	99	99
	VECTO Regional delivery	60	70	61.66
	WHVC	50	50	50
	WHVCx2 Back to back	70	72	70.33
	Fige	40	120	110.33

TABLE 3. SUMMARY OF TAMPERING SCENARIOS, DRIVING CYCLES, AND DETECTION TIME. HIDDEN SIGNAL: UREA DOSAGE COMMAND (A TOTAL OF 15 EXPERIMENTS, WITH DIFFERENT HIDING TECHNIQUES).

Tampering scenario	**Driving Scenario**	**Detection Delay [s]**		
		Min.	**Max.**	**Avg.**
Hidden Urea dosing command signal	VECTO Long haul	535	850	775
	VECTO Regional delivery	640	640	640
	WHVC	600	630	615
	WHVCx2 Back to back	600	600	600
	Fige	300	300	300

aftertreatment simulation model of a heavy-duty vehicle. Datasets which contained known, as well as unknown (e.g., possible future) tampering scenarios. As future work, we intend to further refine and extend the developed tampering detection technique while integrating a prototype within a real automotive environment.

Acknowledgment

This work was funded by the European Union's Horizon 2020 Research and Innovation Programme through DIAS project (*https://dias-project.com/*) under Grant Agreement No. 814951. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] Wu, Zhang, and Qinan Wang. "A single CUSUM chart using a single observation to monitor a variable." International Journal of Production Research 45, no. 3 (2007): 719-741.

[2] Rasch, D., Gibbons, J. D.: "Nonparametric Statistical Inference", 2nd. Ed. Statistics: Textbooks and monographs vol. 65. Marcel Dekker, Inc., New York and Basel 1985, XV, 408 S., 41, 25(34,50 US and Canada). Biom. J., 28: 936-936, 1986. https://doi.org/10.1002/bimj.4710280806.

[3] Kingma, Diederik P and Ba, Jimmy, "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980, 2014.

[4] Minsky, Marvin L and Papert, Seymour A, "Perceptrons", MIT press, Cambridge, 1969.

[5] Rumelhart, David E and Hinton, Geoffrey E and Williams, Ronald J, "Learning internal representations by error propagation", California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[6] Sepp Hochreiter, "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 1998 06:02, 107-116.

[7] Hochreiter, Sepp and Schmidhuber, Jürgen, "Long short-term memory", Neural computation 1997, MIT Press, 1997, 1735–1780.

[8] Jürgen Schmidhuber, Daan Wierstra, Matteo Gagliolo, and Faustino Gomez. 2007. "Training Recurrent Networks by Evolino". Neural Computation 19, 3 (March 2007), 757–779. DOI:https://doi.org/10.1162/neco.2007.19.3.757.

[9] Mozer, Michael, "Induction of Multiscale Temporal Structure", Advances in neural information processing systems, vol. 4, 1991.

[10] Schmidhuber, Jürgen. "Learning complex, extended sequences using the principle of history compression." Neural Computation 4.2 (1992): 234-242.

[11] Kasab, John, and Andrea Strzelec. "Automotive emissions regulations and exhaust aftertreatment systems.", SAE, 2020.

[12] Baldini, Gianmarco, Raimondo Giuliani, and Monica Gemo. "Mitigation of Odometer Fraud for In-Vehicle Security Using the Discrete Hartley Transform." In 2020 11th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), pp. 0479-0485. IEEE, 2020.

[13] Terzi, Sofia, Charalampos Savvaidis, Konstantinos Votis, Dimitrios Tzovaras, and Ioannis Stamelos. "Securing emission data of smart vehicles with blockchain and self-sovereign identities." In 2020 IEEE International Conference on Blockchain (Blockchain), pp. 462-469. IEEE, 2020.

[14] J.A. van den Meiracker and R. Vermeulen, "Deliverable 3.2: Status quo of critical tampering techniques and proposal of required new OBD monitoring functions", DIAS project deliverable: Smart Adaptive Remote Diagnostic Antitampering Systems, (https://www.dias-project.com/Deliverables/All_WPs), December 2020.

[15] J.A. van den Meiracker and R. Vermeulen, "Deliverable 3.1: The market of cheating devices and testing matrix with a prioritization for testing of vehicle tampering technique combinations", DIAS project deliverable: Smart Adaptive Remote Diagnostic Antitampering Systems (https://www.dias-project.com/Deliverables/All_WPs), 2020.

[16] I. Ertug, "Motion for a european parliament solution with recommendations to the Commission on odometer manipulation in motor vehicles: revision of the EU legal framework", Report of the European parliament, Committee on Transport and Tourism, 2018, https://www.europarl.europa.eu/doceo/document/A-8-2018-0155_EN.html.

[17] Ministry of Environment and Food of Denmark, "Measurements of cheating with SCR catalysts on heavy duty vehicles", Environmental Protection Agency, Environmental project No. 2021, 2018, https://www2.mst.dk/Udgiv/publications/2018/06/978-87-93710-42-9.pdf.

[18] Pöhler, Denis, Tim Adler, Chsristopher Krufczik, Martin Horbanski, Johannes Lampel, and Ulrich Platt. "Real driving NOx emissions of European trucks and detection of manipulated emission systems." In EGU General Assembly Conference Abstracts, p. 13991. 2017.

[19] Gadallah, Aly H., Elshenawy A. Elshenawy, Aly M. Elzahaby, Hafez A. El-Salmawy, and Ahmed H. Bawady. "Application of neural networks for prediction and optimization of emissions and performance in a hydrogen fuelled direct injection engine equipped with in cylinder water injection." No. 2009-01-2684. SAE Technical Paper, 2009.

[20] Zhang, Qingning, Andrew Pennycott, Richard Burke, Sam Akehurst, and Chris Brace. "Predicting the nitrogen oxides emissions of a diesel engine using neural networks." No. 2015-01-1626. SAE Technical Paper, 2015.

[21] Anand, G., S. Gopinath, M. R. Ravi, I. N. Kar, and J. P. Subrahmanyam. "Artificial neural networks for prediction of efficiency and NOx emission of a spark ignition engine." No. 2006-01-1113. SAE Technical Paper, 2006.

[22] Arsie, Ivan, Cesare Pianese, and Marco Sorrentino. "Development and real-time implementation of recurrent neural networks for AFR prediction and control." SAE international journal of passenger cars-electronic and electrical systems 1.2008-01-0993 (2008): 403-412.

[23] Desantes, José M., Jose J. Lopez, Jose M. Garcia, and Leonor Hernández. "Application of neural networks for prediction and optimization of exhaust emissions in a HD diesel engine." SAE Transactions (2002): 1993-2002.

[24] Warey, Alok, Jian Gao, and Ronald Grover. "Prediction of Engine-Out Emissions Using Deep Convolutional Neural Networks." SAE International Journal of Advances and Current Practices in Mobility 3, no. 2021-01-0414 (2021): 2863-2871.

[25] Kim, Seil, Aram Cho, and Dong Hoon Lee. "Analysis of Threats and Countermeasures for Odomter Protection." International Journal of Automotive Technology 21, no. 5 (2020): 1271-1281.

[26] Yeon, Kyuhwan, Kyunghan Min, Jaewook Shin, Myoungho Sunwoo, and Manbae Han. "Ego-vehicle speed prediction using a long short-term memory based recurrent neural network." International Journal of Automotive Technology 20, no. 4 (2019): 713-722.

[27] Shin, Jongho, Youngmi Baek, Yongsoon Eun, and Sang Hyuk Son. "Intelligent sensor attack detection and identification for automotive cyber-physical systems." In 2017 IEEE Symposium series on computational intelligence (SSCI), pp. 1-8. IEEE, 2017.

[28] Roman, Adrian-Silviu, Béla Genge, Adrian-Vasile Duka, and Piroska Haller. "Privacy-Preserving Tampering Detection in Automotive Systems." Electronics 10, no. 24, (2021): 3161.

[29] Bolboacă, Roland, Teri Lenard, Béla Genge, and Piroska Haller. "Locality sensitive hashing for tampering detection in automotive systems." In Proceedings of the 15th International Conference on Availability, Reliability and Security, pp. 1-7. 2020.

[30] Stabili, Dario, Mirco Marchetti, and Michele Colajanni. "Detecting attacks to internal vehicle networks through Hamming distance." In 2017 AEIT International Annual Conference, pp. 1-6. IEEE, 2017.

[31] Groza, Bogdan, and Pal-Stefan Murvay. "Efficient intrusion detection with bloom filtering in controller area networks." IEEE Transactions on Information Forensics and Security 14, no. 4 (2018): 1037-1051.

[32] Longari, Stefano, Daniel Humberto Nova Valcarcel, Mattia Zago, Michele Carminati, and Stefano Zanero. "CANnolo: An anomaly detection system based on LSTM autoencoders for controller area network." IEEE Transactions on Network and Service Management 18, no. 2 (2020): 1913-1924.

[33] Exothermia GmbH, "Exothermia suite", https://www.exothermia.com/exhaust-overview.

[34] He, Tianjia, Lin Zhang, Fanxin Kong, and Asif Salekin. "Exploring inherent sensor redundancy for automotive anomaly detection." In 2020 57th ACM/IEEE Design Automation Conference (DAC), pp. 1-6. IEEE, 2020.

[35] World Harmonized Vehicle Cycle (WHVC). URL: https://dieselnet.com/standards/cycles/whvc.php . Last accessed: 24.02.2022.

[36] European Transient Cycle (ETC). URL: https://dieselnet.com/standards/cycles/etc.php. Last accessed: 24.02.2022.

[37] Directive 1999/96/EC of the European Parliament and of the Council of 13 December 1999. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A31999L0096. Last accessed: 24.02.2022.

[38] COMMISSION REGULATION (EU) 2017/2400 of 12 December 2017, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32017R2400from=EN. Last accessed: 24.02.2022.