Enhanced Anomaly Detection in Automotive Systems Using SAAD: Statistical Aggregated Anomaly Detection

1st Dacian Goina
Faculty of Mathematics and Computer Science
West University of Timisoara
Timisoara, Timis
dacian.goina00@e-uvt.ro

2nd Eduard Hogea
Faculty of Mathematics and Computer Science
West University of Timisoara
Timisoara, Timis
eduard.hogea00@e-uvt.ro

3rd George Maties
Faculty of Electrical Engineering
Gheorghe Asachi
Technical University of Iasi
Iasi, Iasi
george.maties@gmail.com

***Abstract*—This paper presents a novel anomaly detection methodology termed Statistical Aggregated Anomaly Detection (SAAD). The SAAD approach integrates advanced statistical techniques with machine learning, and its efficacy is demonstrated through validation on real sensor data from a Hardware-in-the-Loop (HIL) environment within the automotive domain. The key innovation of SAAD lies in its ability to significantly enhance the accuracy and robustness of anomaly detection when combined with Fully Connected Networks (FCNs) augmented by dropout layers. Comprehensive experimental evaluations indicate that the standalone statistical method achieves an accuracy of 72.1%, whereas the deep learning model alone attains an accuracy of 71.5%. In contrast, the aggregated method achieves a superior accuracy of 88.3% and an F1 score of 0.921, thereby outperforming the individual models. These results underscore the effectiveness of SAAD, demonstrating its potential for broad application in various domains, including automotive systems.**

***Index Terms*—anomaly detection, HIL, machine learning, automotive systems, fully connected networks, statistical methods**

I. Introduction

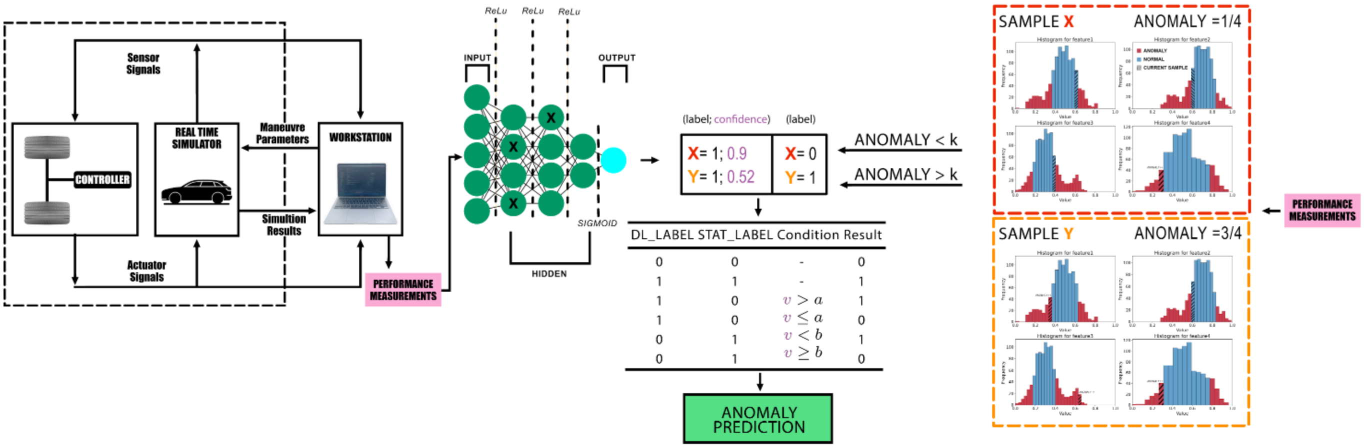
Anomaly detection is a critical task in various fields, particularly in the automotive industry, where it plays a significant role in ensuring the reliability and safety of systems. With the advent of advanced machine learning (ML) techniques, anomaly detection has seen substantial improvements in accuracy and efficiency. Machine learning models, especially deep learning architectures, have shown promising results in identifying abnormal patterns that could indicate potential faults or cyber-attacks in automotive systems [1], [2].

In the context of automotive systems, anomaly detection is crucial for maintaining the operational integrity of both traditional and autonomous vehicles. Autonomous vehicles, in particular, rely heavily on a multitude of sensors to navigate and operate safely. These sensors provide continuous streams of data that need to be monitored in real-time for any signs of anomalies that could compromise the vehicle's safety and functionality [3], [4]. Machine learning models have been employed to analyze these data streams, allowing for the early detection of issues and the implementation of corrective actions before any adverse events occur [5].

To validate and verify the performance of control systems under real-time conditions, Hardware-in-the-Loop (HIL) testing is an essential technique used in the industry. HIL testing involves integrating actual hardware components with simulated environments, providing a highly controlled and repeatable setting to assess the behavior of automotive systems. This method is particularly beneficial for anomaly detection, as it allows for the collection of high-fidelity data that can be used to train and test machine learning models under various scenarios [6]. Additionally, HIL testing helps mitigate the common issue of imbalanced data in anomaly detection datasets by enabling the generation of synthetic data to balance the dataset.

Evaluating anomaly detection algorithms often requires specific metrics tailored for time series data, as classical precision and recall metrics may not capture the nuanced performance characteristics required for detecting anomalies in continuous data streams [7]. Advanced evaluation metrics consider the duration and timing of anomalies, providing a more robust assessment of an algorithm's performance in real-world conditions.

Recent studies have explored various machine learning approaches for anomaly detection in different domains. For instance, [8] utilized explainable random forests to detect anomalies in oil wells, achieving high accuracy and demonstrating the effectiveness of explainable AI in critical applications. Another study [9] proposed a clustered deep one-class classification model for detecting anomalies in cyber-physical systems, showing significant performance improvements over traditional models. A survey on anomaly detection in IoT time-series data highlighted the unique challenges and proposed solutions for implementing these techniques in rapidly expanding fields like the Internet of Things [10]. Additionally, a comprehensive review on outlier detection in time series data provided a structured overview of unsupervised techniques and their applications [11].



DL_LABEL	STAT_LABEL	Condition	Result
0	0	-	0
1	1	-	1
1	0	$v > a$	1
1	0	$v \leq a$	0
0	1	$v < b$	1
0	1	$v \geq b$	0

Fig. 1. Architecture of the SAAD(Statistical Aggregated Anomaly Detection) framework, showcasing the integration of statistical anomaly detection with a Fully Connected Network (FCN) for enhanced anomaly detection in automotive systems. The architecture includes data preprocessing, artificial labeling, deep learning model training, and result aggregation to achieve improved accuracy and robustness. The effectiveness of SAAD is demonstrated with two samples, X and Y. The FCN initially predicts that both are anomalies, with sample X having high confidence and sample Y having low confidence. The statistical method, using $k = 2$, detects sample X as an anomaly and sample Y as not. Aggregation is applied to obtain the final answer, depending on the outputs of the models and the confidence of the deep learning one, combining the strengths of both methods.

The integration of ML in the industry has also yielded impressive results in structural health monitoring [12], [13]. Building on these advancements, recent studies have expanded the field towards hybrid-assisted models that leverage logical rules. These neuro-symbolic approaches combine deep learning with knowledge-based systems to enhance performance. Research has shown that these hybrid models, which incorporate domain-specific knowledge, significantly outperform standard deep learning models by providing more accurate and explainable results [14], [15]. Additionally, innovative methods such as GraphLIME have demonstrated significant improvements in model interpretability and performance in predictive tasks, further showcasing the potential of combining explainable AI techniques with deep learning [16]. The interpretability aspect is highly valuable, as seen in [17], where the insights were used to understand and solve the feature distribution issue that occurred with the generated pseudo-features. A comprehensive comparison in [18] highlights both the advantages of hybrid models and the challenges of creating and scaling them. These hybrid approaches, which synergize complementary components, form the cornerstone of the conceptual framework presented in this paper.

In this paper, we introduce Statistical Aggregated Anomaly Detection (SAAD), a novel anomaly detection method designed to effectively integrate statistical techniques for outlier detection with the performance of supervised machine learning models. Our approach specifically aims to enhance the accuracy and robustness of Fully Connected Networks (FCNs) in anomaly detection tasks. Utilizing real-world data collected from Hardware-in-the-Loop (HIL) environments, we demonstrate the effectiveness of SAAD in identifying anomalies with high precision and recall. Additionally, we illustrate how our statistical method can augment the performance of FCNs, rendering them more reliable for real-time applications in automotive systems

II. Overview

While our primary contribution is aggregation of the statistical method with a Fully Connected Network (FCN), the former is also a novel method. This method has been conceptualized by the first author and has been thoroughly adapted to an industrial setting with the help of the second author, proving to produce reliable results on its own. However, its full potential is realized when used in conjunction with a deep learning model. The deep learning model makes predictions and provides confidence scores. In cases where the confidence is low, the statistical method is applied to aggregate the predictions and produce a final, more accurate output.

The contributions of each author for the work presented in this paper are as follows:

- **First author:** The first author conceptualized the statistical approach. He is also responsible for the creation of the artificial labeling, development of the aggregation function for labels, and conduction of the statistical method's experiments.
- **Second author:** The second author played a crucial role in transitioning the theoretical method to an industrial application. Working with the baseline method provided by the third author. Introduced the idea of SAAD and helped showcase the results in this paper.
- **Third author:** Provided the thoroughly tested baseline model. The author also contributed by supplying and generating on demand the necessary data for extensive experiments.

III. Related Works

The field of anomaly detection has garnered significant attention due to the increasing complexity and connectivity of modern systems. Several approaches have been proposed to address the challenges associated with identifying anomalies in various domains, with our particular interest in this paper being the combination of machine learning and statistical methods.

Intrusion Detection Systems (IDS) for networks, particularly the Controller Area Network (CAN) bus in automotive systems, have been extensively reviewed and developed. Lokman et al. [19] provide a comprehensive overview of IDS implementation in the CAN bus system, discussing various detection approaches, deployment strategies, and technical challenges. They highlight the use of hybrid methods that combine frequency-based, machine learning-based, and statistical-based approaches for anomaly detection, emphasizing the importance of a holistic approach to securing networks.

Statistical learning methods have also been applied to anomaly detection in manufacturing environments, which share similarities with other systems in terms of operational variability and data characteristics. Pittino et al. [20] demonstrate the effectiveness of integrating statistical machine learning methods to control charts in detecting anomalies in production machines. Their hybrid approach, designed to function without continuous recalibration, achieves high recall and robustness, showcasing the potential for similar methods in various applications.

Deep learning techniques, particularly those involving Long Short-Term Memory (LSTM) networks, have shown promise in handling the complex and unpredictable nature of time-series data from multiple sensors. Malhotra et al. [21] propose an LSTM-based Encoder-Decoder scheme for anomaly detection, which learns to reconstruct normal time-series behavior and uses reconstruction error to identify anomalies. This method, while primarily focused on machine learning, can benefit from statistical preprocessing to enhance its effectiveness across various types of time-series data, making it relevant for numerous applications.

The combination of statistical methods with machine learning has been explored in various fields, providing valuable insights for their application in anomaly detection. Holloway and Mengersen [22] review the use of statistical machine learning methods in remote sensing, highlighting their relevance for sustainable development goals. Similarly, Amin et al. [23] demonstrate the integration of statistical methods with machine learning for brain tumor detection, achieving high accuracy through a combination of denoising techniques and feature fusion.

ArunKumar et al. [24] utilize Auto-Regressive Integrated Moving Average (ARIMA) and Seasonal Auto-Regressive Integrated Moving Average (SARIMA) models to forecast COVID-19 cases, underscoring the importance of combining statistical modeling with machine learning for accurate predictions. Banitalebi Dehkordi et al. [25] apply a hybrid approach to detect DDoS attacks in software-defined networks (SDN), integrating entropy-based methods with machine learning classifiers to enhance detection accuracy.

In the context of cryptocurrency price prediction, Khedr et al. [26] review traditional statistical and machine learning techniques, demonstrating the effectiveness of machine learning in handling the dynamic and non-seasonal nature of cryptocurrency data. Their work highlights the potential for similar hybrid approaches in anomaly detection.

A structured approach to anomaly detection for in-vehicle networks is presented by Muter et al. [27]. Their work focuses on a systematic methodology for identifying anomalies in vehicular communications, integrating statistical analysis with machine learning-based detection mechanisms to improve accuracy and robustness.

Research Gap and Our Solution. Despite the significant advancements in anomaly detection, there remains a gap in effectively integrating statistical methods with machine learning models to achieve high accuracy and robustness, particularly in real-time applications. Many existing methods either rely heavily on machine learning techniques, which can be data-intensive and computationally expensive, or on statistical methods that may not fully capture the complexity of the data.

Our response to that is SAAD, which effectively demonstrates that standalone models, which perform similarly when used individually, benefit greatly from combining their results. By integrating statistical techniques with Fully Connected Networks (FCNs), SAAD enhances the detection of anomalies by leveraging the strengths of both approaches. Statistical preprocessing improves data quality, while dropout layers in FCNs prevent overfitting, leading to a more robust and accurate anomaly detection system.

IV. Solution methodology

In this section, we detail the comprehensive methodology employed in the development and validation of the SAAD framework. The methodology encompasses three main stages: (1) the artificial labeling procedure, (2) the development of the deep learning model, and (3) the aggregation of results from both methods to achieve enhanced anomaly detection performance. For a visual representation of the architecture of SAAD, we provide Figure 1.

A. *Artificial labeling procedure*

This stage primarily involves assigning artificial labels to dataset instances. This approach is rooted in the fact that most anomaly detection problems deal with unlabeled data. Typically, anomaly detection is an unsupervised learning task due to the difficulty in defining anomalies and the labor-intensive process of manually labeling data, which requires domain expert knowledge. Starting with the initial dataset, but excluding the target column, statistical methods are used to assign artificial or synthetic labels to the data. This method offers a different perspective compared to the supervised usage in deep learning, making their combination complementary. The term “artificial” indicates that the labels generated in this stage are not necessarily the ground-truth labels.

Anomalous measurements are identified for each feature of the dataset by assessing the occurrence probabilities of the values. The concept originates from the fact that anomalies are often not precisely defined; rather, they are described as

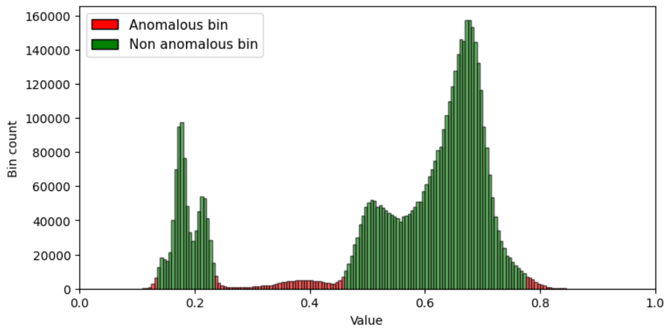


Fig. 2. Anomalous bins exemplification on histogram

instances that exhibit the common behavior [28]. Following this idea, in statistical terms, a value could be considered as being an anomaly if its occurrence probability is lower than a given threshold value. Since it is hard to work directly on high-dimensional data, each feature is examined separately. The occurrence probability of a given value is determined using the probability distribution function (PDF). Determining the exact probability density function (PDF) is often challenging, making it more practical to estimate the PDF using a histogram-based method. A histogram is created by splitting the initial collection of numerical values into groups called bins. Each bin has a defined lower and upper bound, which are used to determine where each value should be placed. The number and widths of the bins could be selected using different methods, but a proper width is determined using the Freedman–Diaconis rule [29]. The number of bins is inferred from the usage of bins with the same width.

The number of values from a given bin is called bin count. The relative frequency of a bin is calculated by dividing the bin count by the total count of all bins. Consider a bin being represented as $b_i = \{(a_{i-1}, a_i),\ c_i,\ m_i\}$, $i \geq 1$: a_{i-1} and a_i are the bin's lower and upper bounds, c_i is the bin count and m_i is the relative frequency. Bins with high counts contain significantly more values, indicating that these values are more common within the initial collection. Consequently, the values in high-count bins have higher occurrence probabilities compared to those in low-count bins. A value that belongs to a bin with a low count could be considered as being an anomaly because it is not so common, i.e., it has a low number of occurrences. A bin b_i is called anomalous if its relative frequency value is lower than a given threshold value t, i.e. $m_i < t$. An exemplification of non-anomalous and anomalous bins on the histogram is presented in fig. 2. Further, a value $x \in \mathbb{R}$ is considered as being an anomaly if it belongs to an anomalous bin.

The selection of anomalous bins is presented in Algorithm 1. B is a collection of collections: each collection B_j contains the histogram bins for the values of feature j. The collection A is used to retain the anomalous bins for each feature and is structured in the same way as B.

Algorithm 1 Anomalous bins selection for all features

```
Require: dataset $D \subseteq \mathbb{R}^{m \times n}$, $t \in [0,1]$, bin width $k$
    (optional, use 'auto' for Freedman–Diaconis rule)
Ensure: $B = \{B_0, B_1, ..., B_{n-1}\}$, $A = \{A_0, A_1, ..., A_{n-1}\}$

  $A$, $B \leftarrow [\ ], [\ ]$ {empty lists}
  $j \leftarrow 0$
  for $j \leftarrow 0$ to $n-1$ do {for each feature}
    $D^j \leftarrow$ values of feature j from all instances
    $B_j \leftarrow get\_bins(D^j,\ k)$
    add $B_j$ to $B$
    $i \leftarrow 0$
    $A_j \leftarrow [\ ]$ {empty list for anomalous bins}
    for $i \leftarrow 0$ to $|B_j| - 1$ do {for each bin}
      $b_i \leftarrow B_{j,i}$ {get current bin}
      if $m_i < t$ then {compare relative freq. with threshold}
        add $b_i$ to $A_j$
      end if
    end for
    add $A_j$ to $A$

  end for

  return $B$, $A$
```

After the selection of anomalous bins for each feature, an instance-level anomaly is defined. An instance is considered an anomaly (label value 1) if it contains at least k anomalous values (features). The procedure is presented in Algorithm 2.

Algorithm 2 Artificial labeling procedure

```
Require: instance $d \in \mathbb{R}^{1 \times n}$, $B = \{B_0, B_1, ..., B_{n-1}\}$, $A =$
    $\{A_0, A_1, ..., A_{n-1}\}$, $k < n$
Ensure: 1 if $d$ is an (artificial) anomaly, 0 otherwise
  $count \leftarrow 0$
  for $j \leftarrow 0$ to $n-1$ do
    $b \leftarrow \texttt{bin\_of\_value}(d_j, B_j)$
    if $b$ in $A_j$ then
      $count \leftarrow count + 1$
    end if
  end for
  if $count \geq k$ then
    return 1
  end if
  return 0
```

B. Deep Learning Model

The data preparation process is crucial for providing the model with clean and well-structured input data. Initially, data is loaded from CSV files located in a specified directory, generating file paths and reading the CSV files. During loading, the data undergoes cleaning to handle missing or undefined values, ensuring the dataset is free from any irregularities that could affect model training. Relevant features, or signals, are then selected from the dataset based on the specified configuration, focusing on the necessary columns for model training. Categorical features have been encoded and subsequently, the data is labeled based on the presence of anomalies, with a binary label (0 or 1) assigned to each data point to indicate whether it is an anomaly.

After labeling, the data is scaled using a standard scaler to normalize the feature values. This normalization step helps in speeding up the convergence during training by ensuring that all features contribute equally to the learning process.

The dense network architecture comprises several fully connected (dense) layers with dropout layers between them to prevent overfitting. The input layer uses the selected relevant features and feeds them to the hidden layers, which consist of multiple dense layers with ReLU activation functions and L2 regularization to mitigate overfitting. The output layer is a single neuron with a sigmoid activation function to produce a binary output indicating the presence or absence of an anomaly.

The model is compiled using the Adam optimizer, a binary cross entropy loss function, and various evaluation metrics including binary accuracy, precision, recall, and custom F-beta scores.

During inference, the model processes new input data in the same manner, outputting a prediction along with a confidence score. The prediction is binary, indicating whether the input data point is an anomaly, while the confidence score reflects the model's certainty about this prediction. The confidence score is particularly important in applying SAAD, as the aggregation of results from the deep learning model and the statistical model is based on the confidence from the former.

C. Aggregated solution

The results provided by the artificial labeling procedure and the deep learning model are aggregated to create a new set of labels. These final results are compared with the ground-truth labels to compute the performance metrics for the proposed approach. For an instance $d \in D$, consider STAT_LABEL as the label (prediction result) provided by the artificial labeling procedure and DL_LABEL as the label provided by the deep learning model for the evaluation of d. For an instance d evaluated by the deep learning model, the outcome value DL_LABEL (either 1 or 0) depends on the value v, computed at the end of the evaluation using the sigmoid function. The value v is further used by the deep learning model to decide the outcome for the evaluated instance d: if, $v > 0.5$ then, the model classifies d as an anomaly (label value 1), otherwise as a non-anomalous instance (label value 0). The aggregation function decides the final label value using the values of STAT_LABEL, DL_LABEL, and v. The decision cases are described below:

STAT_LABEL	DL_LABEL	Condition	Result
0	0	-	0
1	1	-	1
0	1	$v > a$	1
0	1	$v \leq a$	0
1	0	$v < b$	1
1	0	$v \geq b$	0

TABLE I
DECISION RULES FOR LABEL AGGREGATION

V. EXPERIMENT

A. Data Acquisition

The data acquisition process for this study focuses on measuring and collecting data from the Hardware-in-the-Loop (HIL) environment, specifically tailored for underbrake scenarios. This is also presented in the Figure 1. The workbench setup includes several components to simulate and capture the necessary data:

Simulation Workbench. The simulation workbench connects a real-time simulation environment to the physical Electronic Control Unit (ECU) of a vehicle. The simulation environment encompasses detailed models of various vehicle systems such as the engine, transmission, suspension, and steering. These models interact with sensor readings, control signals, and actuator commands to mimic real-world driving conditions.

The HIL setup provides inputs to the ECU, which then processes these inputs using its control algorithms. The vehicle's response, including feedback from sensors such as wheel speed, brake pressure, and brake pad temperature is sent back to the simulation environment. A benefit of this feedback loop is that it allows engineers to test and validate the performance of the vehicle's braking system and identify any unwanted behavior in the ECU's algorithms without the need for physical testing.

Data Acquisition System (DAQ). The DAQ system is a critical component of the workbench, designed to measure and collect high-fidelity data from the HIL environment. The DAQ system comprises the following elements:

- **Sensors**: Devices that measure various parameters during the simulation, including temperature, position, voltage, and other relevant metrics.
- **Data Acquisition Hardware**: This hardware interfaces the sensors with the computer and typically includes an analog-to-digital converter to transform the analog signals from the sensors into digital signals that can be processed by the computer.
- **Software**: The DAQ software is used to configure the hardware, process the collected data, and display the results. It includes drivers necessary for data acquisition and tools for data analysis.
- **Computer System**: A robust computer system is utilized to run the DAQ software and store the collected data for subsequent analysis.

TABLE II
ACCURACY SCORES OBTAINED ON AGGREGATED RESULTS (IN %)

$b \backslash a$	**0.51**	**0.52**	**0.53**	**0.54**	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**	**0.96**	**0.97**	**0.98**	**0.99**	**1**
0.00	54.0	54.3	54.6	54.8	54.9	56.2	59.1	59.9	61.1	62.2	63.4	65.4	68.3	73.0	77.1	77.1	77.1	77.1
0.01	61.8	62.0	62.3	62.5	62.7	63.9	66.9	67.6	68.8	69.9	71.2	73.2	76.0	80.7	84.8	84.8	84.8	84.8
0.02	62.1	62.4	62.7	62.9	63.1	64.3	67.2	68.0	69.2	70.3	71.5	73.5	76.4	81.1	85.2	85.2	85.2	85.2
0.03	62.4	62.7	62.9	63.1	63.3	64.6	67.5	68.3	69.4	70.6	71.8	73.8	76.6	81.3	85.4	85.4	85.4	85.4
0.04	62.6	62.8	63.1	63.3	63.5	64.7	67.6	68.4	69.6	70.7	71.9	74.0	76.8	81.5	85.6	85.6	85.6	85.6
0.05	62.7	63.0	63.2	63.4	63.6	64.8	67.7	68.5	69.7	70.8	72.1	74.1	76.9	81.6	85.7	85.7	85.7	85.7
0.10	63.8	64.0	64.3	64.5	64.7	65.9	68.8	69.6	70.7	71.1	72.4	74.4	77.2	81.9	86.0	86.0	86.0	86.0
0.15	64.4	64.7	64.9	65.1	65.3	66.5	69.4	70.2	71.3	71.4	72.6	74.6	77.4	82.1	86.2	86.2	86.2	86.2
0.20	64.8	65.0	65.3	65.5	65.7	66.9	69.8	70.6	71.7	71.6	72.8	74.8	77.6	82.3	86.4	86.4	86.4	86.4
0.25	65.1	65.4	65.6	65.8	66.0	67.2	70.1	70.9	72.0	71.7	72.9	75.0	77.8	82.5	86.6	86.6	86.6	86.6
0.30	65.4	65.7	65.9	66.1	66.3	67.5	70.4	71.2	72.3	71.8	73.1	75.1	77.9	82.6	86.7	86.7	86.7	86.7
0.35	65.8	66.0	66.3	66.5	66.7	67.9	70.8	71.6	72.7	72.0	73.3	75.3	78.2	82.9	86.9	86.9	86.9	86.9
0.40	66.2	66.4	66.7	66.9	67.1	68.3	71.2	72.0	73.1	72.1	73.3	75.3	78.2	82.9	86.9	86.9	86.9	86.9
0.45	66.6	66.8	67.1	67.3	67.5	68.7	71.6	72.4	73.5	72.2	73.4	75.4	78.3	83.0	87.1	87.1	87.1	87.1
0.46	66.9	67.1	67.4	67.6	67.8	69.0	71.9	72.7	73.8	72.6	73.8	75.8	78.7	83.4	87.5	87.5	87.5	87.5
0.47	67.1	67.4	67.6	67.8	68.0	69.2	72.1	72.9	74.0	72.8	74.0	76.0	78.9	83.6	87.7	87.7	87.7	87.7
0.48	67.3	67.6	67.8	68.0	68.2	69.4	72.3	73.1	74.2	73.0	74.2	76.2	79.1	83.8	87.9	87.9	87.9	87.9
0.49	67.6	67.8	68.1	68.3	68.5	69.7	72.6	73.4	74.5	73.2	74.4	76.4	79.3	84.0	88.1	88.1	88.1	88.1
0.50	67.8	68.0	68.3	68.5	68.7	69.9	72.8	73.6	74.7	73.4	74.6	76.6	79.5	84.2	88.3	88.3	88.3	88.3

TABLE III
F1 SCORES OBTAINED ON AGGREGATED RESULTS

$b \backslash a$	**0.51**	**0.52**	**0.53**	**0.54**	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**	**0.96**	**0.97**	**0.98**	**0.99**	**1**
0.00	0.52	0.53	0.53	0.54	0.54	0.56	0.60	0.61	0.63	0.65	0.66	0.69	0.72	0.78	0.82	0.82	0.82	0.82
0.01	0.63	0.64	0.64	0.64	0.65	0.66	0.70	0.71	0.72	0.74	0.75	0.78	0.81	0.85	0.89	0.89	0.89	0.89
0.02	0.64	0.64	0.65	0.65	0.65	0.67	0.70	0.71	0.73	0.74	0.76	0.78	0.81	0.85	0.89	0.89	0.89	0.89
0.03	0.64	0.65	0.65	0.65	0.66	0.67	0.71	0.72	0.73	0.75	0.76	0.78	0.81	0.86	0.89	0.89	0.89	0.89
0.04	0.65	0.65	0.65	0.66	0.66	0.67	0.71	0.72	0.73	0.75	0.76	0.78	0.81	0.86	0.89	0.89	0.89	0.89
0.05	0.65	0.65	0.65	0.66	0.66	0.68	0.71	0.72	0.74	0.75	0.76	0.79	0.81	0.86	0.89	0.89	0.89	0.89
0.10	0.66	0.66	0.67	0.67	0.67	0.69	0.72	0.73	0.75	0.75	0.77	0.79	0.82	0.86	0.90	0.90	0.90	0.90
0.15	0.67	0.67	0.67	0.68	0.68	0.69	0.73	0.74	0.75	0.75	0.77	0.79	0.82	0.86	0.90	0.90	0.90	0.90
0.20	0.67	0.67	0.68	0.68	0.68	0.70	0.73	0.74	0.76	0.76	0.77	0.79	0.82	0.87	0.90	0.90	0.90	0.90
0.25	0.68	0.68	0.68	0.68	0.69	0.70	0.74	0.75	0.76	0.76	0.77	0.79	0.82	0.87	0.90	0.90	0.90	0.90
0.30	0.68	0.68	0.68	0.69	0.69	0.71	0.74	0.75	0.77	0.76	0.77	0.80	0.82	0.87	0.90	0.90	0.90	0.90
0.35	0.68	0.68	0.69	0.69	0.69	0.71	0.74	0.76	0.77	0.76	0.78	0.80	0.83	0.87	0.91	0.91	0.91	0.91
0.40	0.68	0.69	0.69	0.69	0.70	0.71	0.75	0.76	0.77	0.76	0.78	0.80	0.83	0.87	0.91	0.91	0.91	0.91
0.45	0.69	0.69	0.69	0.70	0.70	0.72	0.75	0.76	0.78	0.76	0.78	0.80	0.83	0.87	0.91	0.91	0.91	0.91
0.46	0.69	0.69	0.70	0.70	0.70	0.72	0.75	0.76	0.78	0.77	0.78	0.80	0.83	0.87	0.91	0.91	0.91	0.91
0.47	0.69	0.70	0.70	0.70	0.70	0.72	0.76	0.77	0.78	0.77	0.78	0.80	0.83	0.88	0.91	0.91	0.91	0.91
0.48	0.70	0.70	0.70	0.70	0.71	0.72	0.76	0.77	0.78	0.77	0.79	0.81	0.84	0.88	0.92	0.92	0.92	0.92
0.49	0.70	0.70	0.71	0.71	0.71	0.73	0.76	0.77	0.79	0.77	0.79	0.81	0.84	0.88	0.92	0.92	0.92	0.92
0.50	0.70	0.71	0.71	0.71	0.71	0.73	0.77	0.78	0.79	0.78	0.79	0.81	0.84	0.89	0.92	0.92	0.92	0.92

Data Collection Process. The raw data is represented by measurements obtained from the ECU within the HIL environment. Each measurement consists of thousands of signals sampled at 10ms intervals, with each signal potentially representing a unique feature.

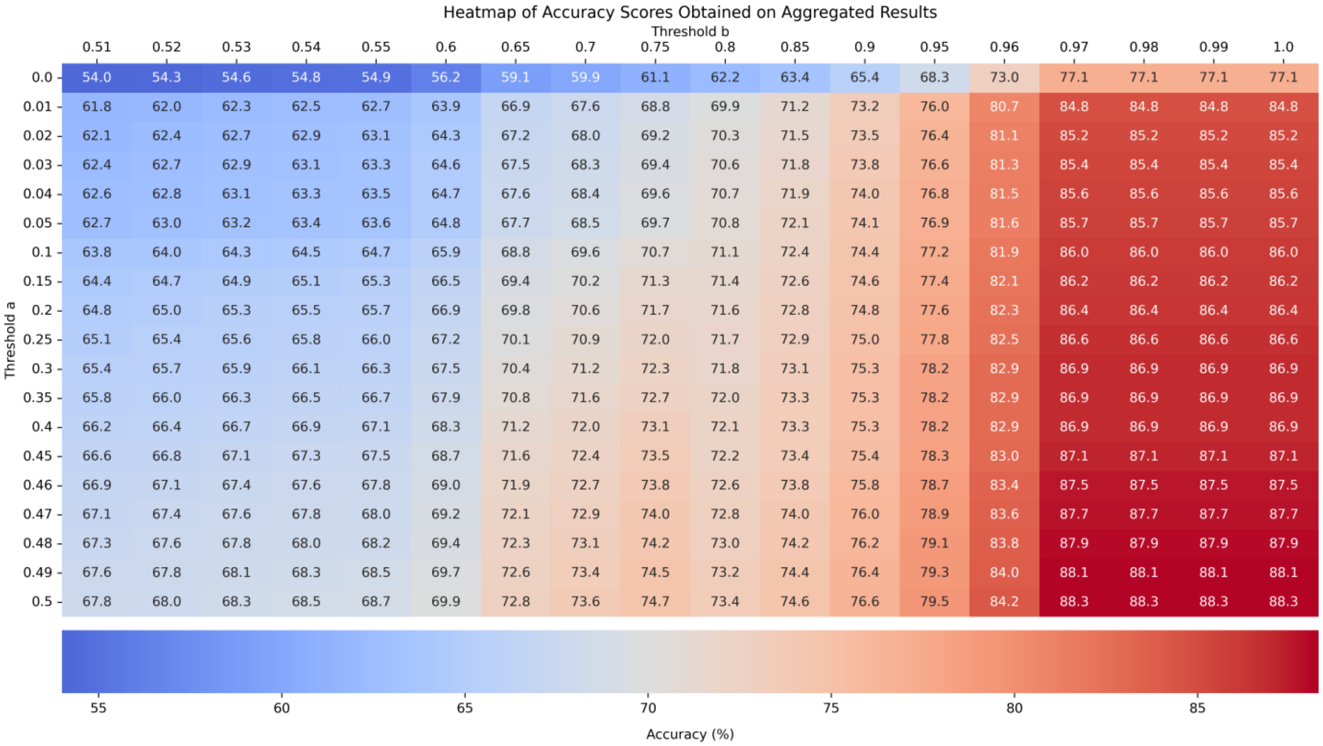
To gather these measurements, various vehicle maneuvers were simulated to reproduce underbrake scenarios. Key variables for these maneuvers included vehicle speed (ranging from 0 to 150 km/h) and brake pedal pressure (ranging from 5% to 100%). These maneuvers ensured a diverse and representative dataset, which is essential for training robust machine learning models.

Experimental Setup. For this study, a total of 150 vehicle maneuvers were simulated for the training set to capture a wide range of underbrake scenarios. Additionally, distinct maneuvers were created for the testing set, featuring multiple underbrake activations and varying speeds. This approach helps to evaluate the performance of the anomaly detection algorithms and prevents overfitting.

By carefully designing the data acquisition and preprocessing steps, we ensure that the machine learning models trained on this data can make accurate predictions and decisions, ultimately improving the reliability of automotive systems.

B. *Experiment methodology*

As proof of concept, the proposed method was used to detect anomalies on the data collected from a HIL environment: as mentioned in previous section, the used data represent measurements taken from different usages of braking function of vehicles on different situations; the used datasets contain 18 features and the target class.



	0.51	0.52	0.53	0.54	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	0.96	0.97	0.98	0.99	1.0
0.0	54.0	54.3	54.6	54.8	54.9	56.2	59.1	59.9	61.1	62.2	63.4	65.4	68.3	73.0	77.1	77.1	77.1	77.1
0.01	61.8	62.0	62.3	62.5	62.7	63.9	66.9	67.6	68.8	69.9	71.2	73.2	76.0	80.7	84.8	84.8	84.8	84.8
0.02	62.1	62.4	62.7	62.9	63.1	64.3	67.2	68.0	69.2	70.3	71.5	73.5	76.4	81.1	85.2	85.2	85.2	85.2
0.03	62.4	62.7	62.9	63.1	63.3	64.6	67.5	68.3	69.4	70.6	71.8	73.8	76.6	81.3	85.4	85.4	85.4	85.4
0.04	62.6	62.8	63.1	63.3	63.5	64.7	67.6	68.4	69.6	70.7	71.9	74.0	76.8	81.5	85.6	85.6	85.6	85.6
0.05	62.7	63.0	63.2	63.4	63.6	64.8	67.7	68.5	69.7	70.8	72.1	74.1	76.9	81.6	85.7	85.7	85.7	85.7
0.1	63.8	64.0	64.3	64.5	64.7	65.9	68.8	69.6	70.7	71.1	72.4	74.4	77.2	81.9	86.0	86.0	86.0	86.0
0.15	64.4	64.7	64.9	65.1	65.3	66.5	69.4	70.2	71.3	71.4	72.6	74.6	77.4	82.1	86.2	86.2	86.2	86.2
0.2	64.8	65.0	65.3	65.5	65.7	66.9	69.8	70.6	71.7	71.6	72.8	74.8	77.6	82.3	86.4	86.4	86.4	86.4
0.25	65.1	65.4	65.6	65.8	66.0	67.2	70.1	70.9	72.0	71.7	72.9	75.0	77.8	82.5	86.6	86.6	86.6	86.6
0.3	65.4	65.7	65.9	66.1	66.3	67.5	70.4	71.2	72.3	71.8	73.1	75.3	78.2	82.9	86.9	86.9	86.9	86.9
0.35	65.8	66.0	66.3	66.5	66.7	67.9	70.8	71.6	72.7	72.0	73.3	75.3	78.2	82.9	86.9	86.9	86.9	86.9
0.4	66.2	66.4	66.7	66.9	67.1	68.3	71.2	72.0	73.1	72.1	73.3	75.3	78.2	82.9	86.9	86.9	86.9	86.9
0.45	66.6	66.8	67.1	67.3	67.5	68.7	71.6	72.4	73.5	72.2	73.4	75.4	78.3	83.0	87.1	87.1	87.1	87.1
0.46	66.9	67.1	67.4	67.6	67.8	69.0	71.9	72.7	73.8	72.6	73.8	75.8	78.7	83.4	87.5	87.5	87.5	87.5
0.47	67.1	67.4	67.6	67.8	68.0	69.2	72.1	72.9	74.0	72.8	74.0	76.0	78.9	83.6	87.7	87.7	87.7	87.7
0.48	67.3	67.6	67.8	68.0	68.2	69.4	72.3	73.1	74.2	73.0	74.2	76.2	79.1	83.8	87.9	87.9	87.9	87.9
0.49	67.6	67.8	68.1	68.3	68.5	69.7	72.6	73.4	74.5	73.2	74.4	76.4	79.3	84.0	88.1	88.1	88.1	88.1
0.5	67.8	68.0	68.3	68.5	68.7	69.9	72.8	73.6	74.7	73.4	74.6	76.6	79.5	84.2	88.3	88.3	88.3	88.3

Fig. 3. Heatmap of Accuracy Scores Obtained on Aggregated Results. The heatmap visualizes the accuracy percentages across different thresholds a and b. The color intensity represents the accuracy, with warmer colors indicating higher accuracy scores. The pattern shows that accuracy generally increases as both thresholds a and b increase, with the highest accuracy observed in the lower right corner of the heatmap, where both thresholds are at their maximum values. This indicates a positive correlation between higher threshold values and improved accuracy.

Models parameters configuration. In the case of artificial labeling procedure, the used parameters are: number of minimum anomalous values per instance $k = 3$ and anomalous bin relative frequency threshold $t = 0.05$. The artificial labeling method works on an unsupervised way: it assigns labels just by usages of statistical methods, without using ground truth labels. The used parameters values were chosen such that, the percentage of artificial anomalous labels to be closer to the percentage of ground-truth anomalous labels.

For the deep learning model, to further enhance the model's performance, hyperparameter optimization is applied using Bayesian optimization with Keras Tuner. The optimizer defines a search space for several key hyperparameters, including the number of units in each dense layer, learning rate, dropout rates, and regularization parameters.

The optimization process iteratively evaluates different combinations of these hyperparameters by training the model on the training data and validating it on a separate validation set. Early stopping is employed to prevent overfitting, halting training when the model's performance on the validation set ceases to improve. The best-performing model configuration is then selected based on the evaluation metrics.

C. Experiments results

The artificial labeling method alone provided an accuracy score of 72.1% on the testing data. The deep learning model alone provided on accuracy score of 71.5% on the testing data. The labels provided by each method were aggregated according to the presented strategy. The new labels were compared with ground-truth labels and accuracy and F1 score metrics were used to measure the performances: the results are presented in the table II and III.

As can be observed from results from the tables II and III, the proposed approach that combines artificial labeling method with deep learning model provide higher scores than each method alone. Both parameters a and b from aggregation function have an important role on the quality of the results. Higher a and b allow the used approach to provide better results: labels obtained with artificial labeling method together with higher a and b values enforce the aggregation function to select high confidence results provided by deep learning model. The experimental results for different pairs of a, b values can also be seen in Figure 4.

D. Results interpretation

As presented in the table I, the aggregation approach provided higher performances, in comparison with both models alone. As an initial explanation for the linear / uniform increase in the accuracy and F1 score values, this trend is likely related to the use of small steps in the values used in the tuning for aggregation parameters a and b. As explanations related to the higher performances provided by the aggregation strategy,

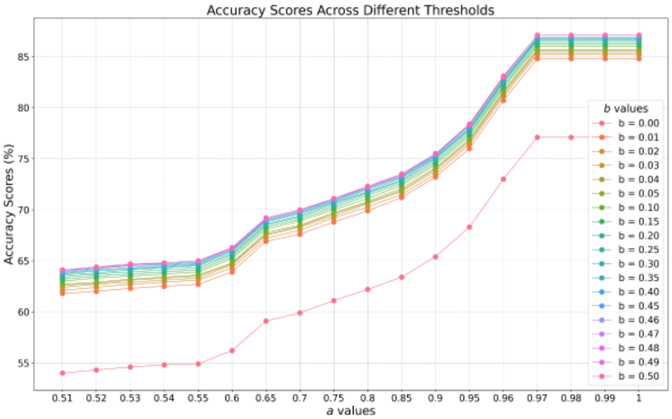


Fig. 4. Evolution of accuracy at different threshold values.

firstly for the case when both models provide the same label, the aggregated label is well understood.

When the deep learning model classifies an instance as an anomaly (label value 1) and the statistical approach classifies it as a normal instance (label value 0), we rely on the deep learning model's result only if it provides a high enough confidence for the given prediction. For example, if $v = 0.55$ this value is close to 0.5, which is the upper bound for what is considered normal. Thus, we use $v > a$ to ensure that the prediction quality is sufficient to consider the evaluated instance as an anomaly. By increasing the value of a, we request a higher confidence v (expect progressively better result), effectively raising the threshold for accepting its predictions. This higher expectation leads to a decrease in the number of instances labeled as anomalies (value 1) provided by the deep learning model. If $a = 1$, the condition $v > 1$ holds nowhere, meaning all instances will be labeled as normal (value 0), relying solely on the statistical approach's labels.

When the deep learning model classifies an instance as non-anomalous (label value 0) and the statistical approach classifies it as an anomaly (label value 1), we rely on the deep learning model if its confidence value v is low enough. A v value lower than 0.5 but still close to it indicates a low confidence level for the evaluated instance to be non-anomalous, so we enforce prediction quality using a constraint rule based on parameter b: $v < b$. In the case of the deep learning model, an instance d is considered as being normal if $v \in [0, 0.5]$; thus, if $b = 0.5$, the condition $v < 0.5$ always holds, thus all such cases receive the label 0 (not anomaly), according to the value provided by the deep learning model.

To reiterate, for $a = 1$ and $b = 0.5$, when the two models provide distinct labels for an evaluated instance, related to the outcomes provided by the aggregation method, all anomaly labels (value 1) are coming from the statistical approach, while all non-anomaly labels (value 0) come from the deep learning model. The best performance is provided by the aggregation function using the parameters $a = 1$ and $b = 0.5$ indicating that the statistical approach performs better at predicting anomalies, whereas the deep learning model excels at predicting non-anomalous instances. An explanation for previous observation relates to the nature of the used models. The deep learning model, due to the unbalanced nature of the dataset (with many more normal instances compared to anomalies), may have a bias favoring the prediction of normal instances over anomalies, leading to better performance in predicting normal instances. Similarly, the statistical approach's superiority in predicting anomalies is likely due to its mode of operation: it determines the anomalous or non-anomalous state of an evaluated instance by examining feature values and checking for low-frequency values. This approach uses several raw thresholds to decide what is an anomaly and what is normal. The statistical approach's outcome can heavily rely on just a few feature values of the evaluated instance. For example, with $k = 4$, means that if the evaluated instance contains at least 4 anomalous values, the instance is classified as an anomaly, even if the rest of the attributes have common values (with respect to the PDF). Therefore, the statistical approach is stricter regarding the values of an instance, making it easier to declare an instance as anomalous: a sufficient number of anomalous values can override the common (non-anomalous) values in the rest of the attributes.

VI. Conclusion

In this paper, we introduced SAAD, a novel statistical anomaly detection method with promising results for automotive systems, validated through extensive testing using real-world data collected from a Hardware-in-the-Loop (HIL) environment. Our main contribution is merging a novel statistical method with deep learning, taking the best of both approaches and achieving higher accuracy with the joint operation than with standalone models.

The results of our experiments demonstrate the efficacy of our proposed method. The standalone statistical method achieved an accuracy of 72.1% on the testing data, while the deep learning model alone provided an accuracy of 71.5%. When combined, the aggregated results yielded an accuracy of up to 88.3% and an F1 score of 0.921.

The integration of statistical techniques with machine learning models highlights the potential for enhancing existing anomaly detection frameworks, making them more reliable and effective in real-world applications.

Future work will focus on further refining the statistical method, exploring adaptive threshold techniques for multivariate time series data, and expanding the application scope to other critical automotive systems. This will help improve the robustness and scalability of our approach, ensuring its effectiveness across diverse and complex scenarios, while also extending its applicability beyond automotive systems.

Acknowledgment

We extend our gratitude to Prof. Darian Onchis for his invaluable guidance and support during the planning stages of this work, and for shaping the direction of our research. His initial works on hybrid models were crucial in recognizing their potential and applicability in the industry.

References

[1] S. A. Jebur, K. A. Hussein, H. K. Hoomod, L. Alzubaidi, and J. Santamaría, "Review on Deep Learning Approaches for Anomaly Event Detection in Video Surveillance," *Electronics*, vol. 12, no. 1, p. 29, Jan. 2023, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2079-9292/12/1/29

[2] A. A. Alsulami, Q. Abu Al-Haija, A. Alqahtani, and R. Alsini, "Symmetrical Simulation Scheme for Anomaly Detection in Autonomous Vehicles Based on LSTM Model," *Symmetry*, vol. 14, no. 7, p. 1450, Jul. 2022, number: 7 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2073-8994/14/7/1450

[3] P. Dixit, P. Bhattacharya, S. Tanwar, and R. Gupta, "Anomaly detection in autonomous electric vehicles using ai techniques: A comprehensive survey," *Expert Systems*, vol. 39, no. 5, p. e12754, 2022.

[4] Q. He, "A machine learning-based anomaly detection framework for connected and autonomous vehicles cyber security," Ph.D. dissertation, University of Nottingham, 2021.

[5] B. Kim, M. A. Alawami, E. Kim, S. Oh, J. Park, and H. Kim, "A Comparative Study of Time Series Anomaly Detection Models for Industrial Control Systems," *Sensors*, vol. 23, no. 3, p. 1310, Jan. 2023, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/23/3/1310

[6] G. Matieş and C. Fosalau, "Detection of abs events in electronic brake systems using machine learning algorithms," in *2023 13th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*. IEEE, 2023, pp. 1–6.

[7] A. Huet, J. M. Navarro, and D. Rossi, "Local Evaluation of Time Series Anomaly Detection Algorithms," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '22. New York, NY, USA: Association for Computing Machinery, Aug. 2022, pp. 635–645. [Online]. Available: https://doi.org/10.1145/3534678.3539339

[8] N. Aslam, I. U. Khan, A. Alansari, M. Alrammah, A. Alghwairy, R. Alqahtani, R. Alqahtani, M. Almushikes, and M. A. Hashim, "Anomaly Detection Using Explainable Random Forest for the Prediction of Undesirable Events in Oil Wells," *Applied Computational Intelligence and Soft Computing*, vol. 2022, p. e1558381, Aug. 2022, publisher: Hindawi. [Online]. Available: https://www.hindawi.com/journals/acisc/2022/1558381/

[9] Y. Kim and H. K. Kim, "Anomaly Detection using Clustered Deep One-Class Classification," in *2020 15th Asia Joint Conference on Information Security (AsiaJCIS)*, Aug. 2020, pp. 151–157. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9194140

[10] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly Detection for IoT Time-Series Data: A Survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020, conference Name: IEEE Internet of Things Journal. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8926446

[11] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A Review on Outlier/Anomaly Detection in Time Series Data," *ACM Computing Surveys*, vol. 54, no. 3, pp. 56:1–56:33, Apr. 2021. [Online]. Available: https://doi.org/10.1145/3444690

[12] H. D. M. Onchis, "A deep learning approach to condition monitoring of cantilever beams via time-frequency extended signatures," *Computers in Industry*, vol. 105, pp. 177–181, Feb. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361518305669

[13] D. M. Onchis and G.-R. Gillich, "Stable and explainable deep learning damage prediction for prismatic cantilever steel beam," *Computers in Industry*, vol. 125, p. 103359, Feb. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361520305935

[14] D. Onchis, C. Istin, and E. Hogea, "A Neuro-Symbolic Classifier with Optimized Satisfiability for Monitoring Security Alerts in Network Traffic," *Applied Sciences*, vol. 12, no. 22, p. 11502, Jan. 2022, number: 22 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-3417/12/22/11502

[15] D. M. Onchis, G.-R. Gillich, E. Hogea, and C. Tufisi, "Neuro-symbolic model for cantilever beams damage detection," *Computers in Industry*, vol. 151, p. 103991, Oct. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361523001410

[16] F. Costi, D. Onchis, E. Hogea, and C. Istin, "Predictive Modeling for Diabetes Using GraphLIME," Mar. 2024, pages: 2024.03.14.24304281. [Online]. Available: https://www.medrxiv.org/content/10.1101/2024.03.14.24304281v1

[17] E. Hogea, A. Popescu, D. Onchis, and G. Petit, "FeTrIL++: Feature Translation for Exemplar-Free Class-Incremental Learning with Hill-Climbing," Mar. 2024, arXiv:2403.07406 [cs]. [Online]. Available: http://arxiv.org/abs/2403.07406

[18] D. M. Onchis, C. Istin, and H. Eduard-Florin, "Advantages of a neuro-symbolic solution for monitoring IT infrastructures alerts," in *2022 24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. Hagenberg / Linz, Austria: IEEE, Sep. 2022, pp. 189–194. [Online]. Available: https://ieeexplore.ieee.org/document/10130897/

[19] S.-F. Lokman, A. T. Othman, and M.-H. Abu-Bakar, "Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review," *J Wireless Com Network*, vol. 2019, no. 1, p. 184, Dec. 2019. [Online]. Available: https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-019-1484-3

[20] F. Pittino, M. Puggl, T. Moldaschl, and C. Hirschl, "Automatic Anomaly Detection on In-Production Manufacturing Machines Using Statistical Learning Methods," *Sensors*, vol. 20, no. 8, p. 2344, Jan. 2020, number: 8 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/20/8/2344

[21] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection," Jul. 2016, arXiv:1607.00148 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1607.00148

[22] J. Holloway and K. Mengersen, "Statistical Machine Learning Methods and Remote Sensing for Sustainable Development Goals: A Review," *Remote Sensing*, vol. 10, no. 9, p. 1365, Sep. 2018, number: 9 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2072-4292/10/9/1365

[23] J. Amin, M. Sharif, M. Raza, T. Saba, and M. A. Anjum, "Brain tumor detection using statistical and machine learning method," *Computer Methods and Programs in Biomedicine*, vol. 177, pp. 69–79, Aug. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260718313786

[24] K. E. ArunKumar, D. V. Kalaga, C. M. Sai Kumar, G. Chilkoor, M. Kawaji, and T. M. Brenza, "Forecasting the dynamics of cumulative COVID-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-Regressive Integrated Moving Average (ARIMA) and Seasonal Auto-Regressive Integrated Moving Average (SARIMA)," *Applied Soft Computing*, vol. 103, p. 107161, May 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621000843

[25] A. Banitalebi Dehkordi, M. Soltanaghaei, and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," *J Supercomput*, vol. 77, no. 3, pp. 2383–2415, Mar. 2021. [Online]. Available: https://doi.org/10.1007/s11227-020-03323-w

[26] A. M. Khedr, I. Arif, P. R. P V, M. El-Bannany, S. M. Alhashmi, and M. Sreedharan, "Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey," *Intelligent Systems in Accounting, Finance and Management*, vol. 28, no. 1, pp. 3–34, 2021, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/isaf.1488. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/isaf.1488

[27] M. Muter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks," in *2010 Sixth International Conference on Information Assurance and Security*. Atlanta, GA, USA: IEEE, Aug. 2010, pp. 92–98. [Online]. Available: http://ieeexplore.ieee.org/document/5604050/

[28] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Muller, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, no. 5, p. 756–795, May 2021. [Online]. Available: http://dx.doi.org/10.1109/JPROC.2021.3052449

[29] D. Freedman and P. Diaconis, "On the histogram as a density estimator: L2 theory," *Probability Theory and Related Fields*, vol. 57, no. 4, pp. 453–476, Dec 1981. [Online]. Available: https://doi.org/10.1007/BF01025868