"GEORGE EMIL PALADE" UNIVERSITY OF MEDICINE, PHARMACY, SCIENCE, AND TECHNOLOGY OF TÂRGU MUREȘ

DOCTORAL SCHOOL OF LETTERS, HUMANITIES AND APPLIED SCIENCES

SCIENTIFIC FIELD: INFORMATICS

PHD THESIS SUMMARY

NONLINEAR SYSTEMS MODELING USING MACHINE LEARNING TECHNIQUES WITH APPLICATIONS IN ANOMALY DETECTION

PhD Candidate:
Roland BOLBOACĂ

Scientific Supervisor:
Prof. Eng. Béla GENGE, PhD

TÂRGU MUREŞ
2024

List of Publications

The evaluation of the author's publications complies with the standards set by CNATDCU (National Council for the Recognition of University Degrees, Diplomas and Certificates), applicable to doctoral students enrolled after October 1, 2018. Rankings are presented based on the classification of conferences[1] and journals[2] in the Computer Science field.

1. **Bolboacă Roland**, Haller Piroska, & Genge Béla (2024, -). *Feature Analysis and Ensemble-based Fault Detection Techniques for Nonlinear Systems* Neural Computing and Applications. [**Decision: Major Revision**]
   Journal Paper, Rank: B, 0 points, source: UEFISCDI 2023[2].

2. **Bolboacă Roland**, Haller Piroska, & Genge Béla (2024, -). *Evaluation Techniques for Long Short-Term Memory Models: Overfitting Analysis and Handling Missing Values.* In the 37th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems. [**Accepted**]
   Conference Paper, Rank: C, 2 points, source: CORE2023[1].

3. **Bolboacă Roland**, & Genge Béla (2023, October). *Unsupervised Outlier Detection in Continuous Nonlinear Systems: Hybrid Approaches with Autoencoders and One-Class SVMs.* In International Conference Interdisciplinarity in Engineering. Cham: Springer International Publishing.
   Conference Paper, Rank: Not Ranked, 1 points, source: CORE[1].

4. **Bolboacă Roland**, & Haller Piroska (2023, March). *Performance Analysis of Long Short-Term Memory Predictive Neural Networks on Time Series Data.* Mathematics, 11(6), 1432.
   Journal Paper, Rank: C, 2 points, source: UEFISCDI 2022 (October)[2]

5. **Bolboacă Roland** (2022, October). *Adaptive ensemble methods for tampering detection in automotive aftertreatment systems.* IEEE Access, 10, 105497-105517.
   Journal Paper, Rank: B, 4 points, source: UEFISCDI 2021[2].

6. **Bolboacă Roland**, Haller Piroska, Kontses Dimitris, Papageorgiou-Koutoulas Alexandros, Doulgeris Stylianos, Zingopis Nikolaos, & Samaras Zissis (2022, June). *Tampering detection for automotive exhaust aftertreatment systems using long*

short-term memory predictive networks. In 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) (pp. 358-367). IEEE.

Conference Paper, Rank: NEW, 0 points, source: CORE2021[1].

7. **Bolboacă Roland**, Lenard Teri, Genge Béla, & Haller Piroska (2020, August). *Locality sensitive hashing for tampering detection in automotive systems.* In Proceedings of the 15th International Conference on Availability, Reliability and Security (pp. 1-7). 9th International Workshop on Cyber Crime.
Conference Paper, Rank C, 1 point, source: CORE2020[1].

8. Lenard Teri, **Bolboacă Roland**, Genge Béla, & Haller Piroska (2020, June). *MixCAN: Mixed and backward-compatible data authentication scheme for controller area networks.* In 2020 IFIP Networking Conference (Networking) (pp. 395-403). IEEE.
Conference Paper, Rank A, 4 points, source: CORE2020[1].

9. Lenard Teri, **Bolboacă Roland**, & Genge Béla (2020, September). *LOKI: A lightweight cryptographic key distribution protocol for controller area networks.* In 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP) (pp. 513-519). IEEE.
Conference Paper, Rank: National, 1 point, source: CORE2020[1].

10. **Bolboacă Roland**, Genge Béla, & Haller Piroska (2019, September). *Using Side-Channels to Detect Abnormal Behavior in Industrial Control Systems.* In 2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP) (pp. 435-441). IEEE.
Conference Paper, Rank C, 2 points, source: CORE2018[1].

Publication scores per rank:

- **Rank A: 4 points**
- **Rank B: 4 points**
- **Rank C: 7 points**
- **Rank D: 2 points**

Total publication score: 17 points.

[1]CORE Conference Ranking Portal. http://portal.core.edu.au/conf-ranks/
[2]UEFISCDI. https://uefiscdi.gov.ro/premierea-rezultatelor-cercetarii-articole

Chapter 1

Introduction

Nonlinear systems are prevalent and extensively utilized across a diverse range of domains. As described by Schoukens and Ljung in [1], any system that deviates from linearity is considered nonlinear. In a general context, nonlinear systems are characterized by the absence of a linear relationship between their inputs and outputs.

The observations generated by these systems are usually in the form of time series. Time series data represent a continuous collection of sequentially measured values and can originate from almost all scientific domains, where system measurements are collected or measured over time. In real-life scenarios, in most cases, time series data originate from live observations or real-time system sensor measurements.

A model can be defined as a simplified representation of the relationship between system inputs and outputs, or as described by Bala et al. [2] as "A substitute of any object or system. A written description of a system is a model that presents one aspect of reality. The simulation model is logically complete and describes the dynamic behavior of the system". Strictly from a nonlinear system modeling perspective, three main approaches are utilized, namely white-box, black-box, and grey-box modeling.

Machine learning is becoming an important technique for black-box modeling as the complexity of data and the need for precise predictions have increased. A specific class of machine learning algorithms, namely neural networks, has been extensively utilized for nonlinear modeling tasks [3]. Among the popular models utilized for continuous nonlinear system modeling, Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Nonlinear Autoregressive Neural Networks with eXternal inputs (NARXNN) have gained popularity for their ability to capture intricate temporal dependencies and adapt to dynamic real-world scenarios [4]. However, standard RNNs suffer from what is known as vanishing and exploding gradient [5]. Furthermore, due to their simple architectures and the issues mentioned above, standard RNNs are unable to efficiently learn long sequences [6]. To address these challenges, advanced models, such as LSTMs, have been developed. These advanced models incorporate gating mechanisms along with concepts such as short- and long-term memory.

In real-life scenarios, these systems may be distributed in large geographical areas, with dispersed sensors that feed the readings. This raises the question if such systems should be modeled using a single self-contained model or utilizing multiple smaller models for each system component or subcomponent. The approach here would be to utilize an ensemble of learners [7].

Generally, any ensemble framework can be viewed and defined using three characteristics that affect its performance. The first is the dependence on the trained baseline models, whether sequential or parallel. The second characteristic is the fusion method, which involves choosing a suitable process for combining outputs of the baseline models using different weight voting approaches or meta-learning methods. The third characteristic is the heterogeneity of the baseline models, whether homogeneous or heterogeneous. Although ensemble methods date back more than two decades, recent works still prove their efficiency to this day [8]. Similarly, in the field of anomaly detection, the efficiency of approaches can be greatly improved by using ensemble methods [9].

Transitioning to the practical application of the thesis, specifically in the field of anomaly detection. As illustrated by Aggarwal et al. [10] three primary methodologies are used to detect anomalies: supervised, unsupervised, and semisupervised. Generally, supervised approaches yield great results but are limited to detecting only learned anomalous patterns. Furthermore, obtaining labeled data for all possible scenarios could be difficult or unrealistic in certain scenarios. As a result, unsupervised anomaly detection techniques represent an interesting research topic.

In continuous time series and data streams, the detection of deviations from normal behavior (e.g., change detection) requires the use of prediction and forecasting models [11]. In such instances, normal behavior can be modeled through various techniques, categorizing anomalies as obvious or subtle deviations from normality [12].

Taking into consideration the previous definitions, it becomes obvious that anomaly detection in continuous time series data originating from nonlinear systems presents numerous challenges. First, modeling the time series, or the generative process of the time series, should be carefully addressed as unreliable and inaccurate models yield poor predictions, which in turn translates to imprecise detection. Second, an appropriate similarity measure must be selected. Last, the anomaly detection technique has to be designed with consideration of the temporal component, the nature of the model's output, and the types of deviations to be detected.

Having established an accurate representation of the nonlinear system, using low-complexity prediction models, the next step involves the design of efficient detectors. These detectors must utilize the output of the predictors to detect deviations from the normal learned behavior. We can argue that such detectors can be designed to identify cumulative deviations using point-by-point approaches or window-based techniques.

These approaches are also identified by Blázquez-García et al. in a recent review of anomaly detection techniques in time series data [13].

In a different direction, the authors of [13] highlight the fact that most of the analyzed anomaly detection techniques focus on detecting parts of the time series that differ substantially from the expected value. However, some anomalous points, or series of points, might resemble the anomaly-free time series, and a methodology of cumulating small deviations over time might provide better results in such scenarios. Furthermore, an interesting unexplored direction might be the one in which outliers might propagate to different variables of a time series. Furthermore, an in-depth analysis of the computational costs of running such techniques in real-time scenarios is still needed.

1.1 Research Objectives

Although the main focus of the thesis is the design of prediction-based unsupervised anomaly detection techniques for time series data originating from nonlinear systems, specific research objectives (**RO**) have been defined to address these challenges.

- **RO1:** Develop an enhanced Long Short-Term Memory prediction model for time series data originating from nonlinear systems.
- **RO2:** Develop a new feature ranking and selection methodology using a sensitivity analysis-based approach. In addition, conduct an in-depth analysis of the prediction model in terms of dealing with missing values and overfitting.
- **RO3:** Develop efficient unsupervised detection methodologies that utilize the output of the prediction models, in the context where the predictors are trained only with anomaly-free observations.
- **RO4:** Develop a new adaptive ensemble of detectors that utilizes a new efficient decision aggregation methodology.
- **RO5:** Implement the proposed solutions in diverse environments (e.g., Python, MATLAB) and embedded systems to measure resource usage on devices with limited capabilities.
- **RO6:** Test and validate the proposed methods using real and synthetic time series datasets originating from nonlinear systems. Additionally, apply the proposed prediction and detection techniques to solve real-world problems.

1.2 Thesis Structure

In addition to this introductory chapter, the current thesis is organized into five chapters.

- **Chapter 2:** The second chapter introduces and defines the important concepts that are utilized throughout the thesis, including historical backgrounds. This chapter also presents relevant and recent studies related to the approaches

presented in the thesis, including nonlinear modeling, machine learning approaches, hyperparameter optimization techniques, and anomaly detection solutions.

- **Chapter 3:** The third chapter introduces the LSTMTF model for nonlinear system modeling. LSTMTF encompasses an enhanced LSTM model with a modified version of the Teacher Forcing algorithm. Additionally, this chapter presents an extensive hyperparameter analysis and a prediction performance comparison with various state-of-the-art models.

  The prediction performance comparison is performed between the proposed LSTMTF model and various other machine learning prediction algorithms including NARXNN, Multi-Layer Perceptrons, Recurrent Neural Networks, Support Vector Machines, AutoRegressive with exogenous input models, and Random Forests.
- **Chapter 4:** The fourth chapter introduces two feature selection methodologies that are utilized as a means to reduce the complexity of the LSTMTF model. Additionally, this chapter addresses other significant issues, including model overfitting and mechanisms to handle missing data.
- **Chapter 5:** The fifth chapter illustrates two directions in the field of anomaly detection, namely tampering and fault detection, together with two proposed ensemble-based detection approaches. The performance of the anomaly detection ensembles introduced in this chapter is compared with numerous state-of-the-art supervised and unsupervised approaches.
- **Chapter 6:** The sixth and final chapter of the thesis presents the main conclusions, the original scientific contributions, possible future direction, research funding, and the author's participation in research projects.

Chapter 2

Related Work

2.1 Machine Learning Modeling, Prediction and Teacher Forcing

Today, machine learning algorithms are the "go-to" for solving a plethora of real-life problems in various domains, from healthcare, finance, and manufacturing all the way to agriculture. On a larger scale, machine learning can be considered an umbrella term that incorporates a wide range of algorithms and models proposed for specific tasks, including medical diagnosis [14], predictive maintenance [15], text authorship attribution [16], and anomaly detection [17]. In the direction of system modeling, machine learning techniques are widely employed to create accurate data-driven models. These approaches are utilized in various domains, including earth sciences, various industrial domains, mathematics and physics, and even user action modeling.

One specific subclass of machine learning includes neural networks. Since the first mention of artificial neurons almost 80 years ago [18], neural networks have constantly evolved and have been widely applied in numerous areas and domains. For nonlinear system modeling various well-known architectures are utilized, including RNNs, LSTMs, and NARXNN [3].

In continuous time series and continuous data streams, where an underlying temporal component is present, the detection of deviations from normal behavior requires the utilization of prediction and forecasting models. However, for neural networks to effectively capture the temporal dependencies and dynamics present in nonlinear systems, which can be described by differential equations, the current inputs alone may not be sufficient. In such instances, the previous outputs or states of the modeled system may provide important information about the behavior of the system and may be necessary for accurate predictions of the subsequent outputs [19]. In such systems, the previous output or system state can be seen as an additional input that helps the neural network capture the behavior and dynamics of the system over time [20].

When modeling the dynamics of nonlinear systems, there exists a notable approach where the incorporation of the prior output or system state is regarded as an additional input, thus enhancing the neural network's ability to effectively capture the temporal behavior and dynamics of the system [20]. This can be achieved through the application of a technique known as Teacher Forcing, as introduced by Williams and Zipser in [21].

In short, the original Teacher Forcing (TF) denotes a training algorithm for RNNs, where during training the output ground truth (e.g., observed) value is fed as an additional input to the model, while during inference the model takes the prediction from the previous time step as an extra input.

In Goodfellow's book [5], TF was also presented as a neural network training technique, applicable to recurrent neural networks that have output to hidden connections. The author continues to state that this technique originates from the maximum likelihood criterion, where during the training phase, the neural network receives the ground truth value of the output as input in the next time step. Moreover, Goodfellow also states that TF is also applicable to models that have hidden-to-hidden connections as well. In this scenario, training is carried out using both TF and backpropagation through time (BPTT) [22].

2.2 Hyperparameter Selection, Model Benchmarking and Feature Analysis

Thomas Brueuel, from Google's research team, studied the behavior and performance of LSTM classifiers in [23]. This study analyses the behavior of LSTMs for different hyperparameters, but also how the choice of nonlinearities affects performance. Among the tested hyperparameters we find the learning rate, number of hidden units, and mini-batch size. The authors focused on digit classification on two popular benchmarking datasets, namely MNIST, which is an isolated digit handwriting classification dataset, and UW3, which is an OCR evaluation database. The results revealed that the performance of LSTM classifiers depends mainly on learning rates, while batching has little to no effect. Softmax training produced better results compared to the least squares approach. Moreover, LSTMs without peephole connections yielded superior performance.

In a more recent study, Siami-Namini et al. [24] analyzed the time series forecasting performance of unidirectional LSTMs and bidirectional LSTMS (BiLSTMs). The authors compared the performance of auto-regressive integrated moving average (ARIMA) models, LSTMs, and BiLSTMs in the context of predicting financial time series data. One interesting aspect of this research is the prediction performance when the time series data are learned in both directions (i.e., past-to-future and future-to-past). Their results showed that BiLSTM's training time was slower, but outperformed the unidirectional LSTM and ARIMA models in terms of prediction accuracy. Nonetheless, the authors

provided no architectural or hyperparameter information about the tested neural networks, or whether the two architectures were trained and tested with the same set of hyperparameters.

2.3 Anomaly Detection in Time Series Data

Anomalies or outliers, as defined by Mehrotra et al. in [25], are "substantial deviations from the norm". Here, norm defines normality, be that a state, a behavior, or some proprieties of the observed system, object, process, or even a person or group of persons. This normality is defined depending on the context and domain where anomaly detection approaches are applied. While in [25] the terms anomaly and outlier are used interchangeably, the authors highlight that in some articles the term anomaly is utilized when discussing processes, and outliers are introduced when discussing data.

The first direction focuses on fault detection. Fault detection is a specific application of anomaly detection focused on identifying deviations from the normal behavior, caused by either sensor failures, component failures, or interventions on the system. Yoon and Macgregor in [26] state that "The purpose of fault detection is to determine the occurrence of an abnormal event in a process". Considering large, sometimes distributed systems, the effects of some faults might propagate between the interconnected subsystems or components of the larger system. This effect is visible in the time series generated by the system.

The second direction is focused on detecting anomalies that are explicitly hidden by a malicious individual, this is further named tampering. Tampering denotes a procedure that alters the system behavior in order to gain particular advantages (e.g., financial, operational). Furthermore, in order for tampering to remain undiscovered, the same malicious person hides the effects of the modifications by injecting false readings that mimic the normal behavior of the system. These false injected readings follow a similar distribution as the ones from the normal operating conditions of the system, and they also mask the real readings that would reflect the effects of tampering in certain system components.

Chapter 3

LSTMTF: Enhanced Long Short-Term Memory for Nonlinear Systems Modeling

The original TF version, as proposed for standard RNN models, uses the previously observed output value as an additional input during training, during inference the model output is looped back as input. This chapter describes an enhanced LSTM model named LSTMTF, which combines the standard LSTM with TF in an innovative way. The LSTMTF model utilizes the previously observed output value as an additional input during both the training and inference procedures. This chapter also describes the LSMTFC model, which denotes a standard LSTM model with the TF algorithm, where the previously observed output value is utilized only during training.

The effects of different hyperparameters are analyzed for the newly introduced LSTMTF, LSTMTFC, and the standard LSTM models, using an empirical approach. The hyperparameters include the input sequence length, the number of time delays, the mini-batch size, the learning rate, and the number of hidden units. The models are trained and tested on time series data originating from a well-known nonlinear system, namely on the Tennessee Eastman process dataset [27, 28].

Next, the prediction performances of different variants of the LSTM model are compared with the NARXNN model. The authors of some large-scale studies, such as those in [24], performed thousands of experiments in this direction. Similarly, for this chapter alone, approximately 100,000 models were trained and tested using a wide range of hyperparameters, configurations, and prediction modes.

This chapter also studies the exposure bias effect [29] for neural networks trained with TF in its original form. Although exposure bias can affect the prediction performance of neural networks, this chapter will also discuss the advantages gained by using LSTMs with TF for anomaly detection tasks.

Additionally, to further validate the LSTMTF model, additional experiments are performed. Here, the model is compared with 11 state-of-the-art parametric and nonparametric forecasting, regression, and prediction models.

3.1 Proposed Architecture

3.1.1 Long Short-Term Memory Model

The LSTM model can be defined as an enhanced version of an RNN, capable of capturing long- and short-term dependencies from data sequences, while also solving the exploding and vanishing gradient problem [30]. The standard LSTM model will further be named Vanilla LSTM (VLSTM). This model can comprise multiple layers that incorporate sequential LSTM units. These units take the current input vector, denoted as $X(t)$, together with the previous hidden state vector of the layer, further denoted as $h(t)$. The cell state vector, denoted as $C(t)$, is updated using three gates, namely, the input gate $I(t)$, the forget gate $f(t)$, and the output gate $o(t)$.

3.1.2 Long Short-Term Memory Model with Teacher Forcing

In short, the standard LSTM represents a nonlinear function of the previous inputs and hidden states. Subsequently, the LSTMTF represents a nonlinear function of the previous inputs, hidden states, and observed outputs.

As mentioned above, applying TF involves modifying the training procedures by adding, at each time step, an extra input, which is the previous ground truth value $y(t-1)$. This value is further propagated to all the LSTM gates. This extra input is required during both training and inference. If in the inference phase the output ground truth value is not available, it is replaced by the previous predicted value. TF is applicable to models that have a recurrent connection from their output leading back into the model and can be used as an alternative to Back Propagation Through Time (BPTT) when the model lacks hidden-to-hidden connections. However, TF can still be applied in conjunction with BPTT for training models with hidden-to-hidden connections [5].

As described in [5], training models with the original version of TF can lead to poor prediction results, as during inference the model could be exposed to different data. This is referred to as exposure bias. Exposure bias occurs when a machine learning model is not exposed to a sufficiently diverse range of data during training. Essentially, exposure bias occurs when the distribution of data seen by the model during training does not accurately reflect the distribution of data it will encounter in the real world.

3.2 Hyperparameter Analysis

The performance and predictive capabilities of the three models (i.e., VLSTM, LSTMTF, and LSTMTFC) are tested using two distinct configurations and in two operating modes.

First, in terms of the number of inputs and outputs, the models are tested as multi-input single-output (MISO) and multi-input multi-output (MIMO). The MISO configuration involves predicting one variable using multiple input variables. Conversely, the MIMO configuration involves predicting multiple variables using multiple input variables.

Second, in terms of prediction modes, two approaches are followed, namely, M2O and M2M. In the case of M2O, the models will take as input a sequence of X_n^{φ} inputs; here, φ denotes the number of time steps (in other words, the length of the input sequence) and n denotes the number of input variables and will output only the final predicted value of the sequence. To exemplify, for every sequence of ten input values, the model will output the next value in the sequence.

In the case of M2M, the models similarly take as input a sequence of X_n^{φ} values and output another sequence of values of size φ, the first prediction starting at $t_0 + 1$. Here, t_0 denotes the time of the first value in the input sequence.

3.3 Summary

This chapter introduced the LSTMTF, an LSTM model combined with a new variant of the TF algorithm applied during both training and inference. This chapter also offered an in-depth analysis of LSTM models trained with and without TF. TF was applied in two variants, with the actual (observed) value fed back as input during both training and testing, as proposed for anomaly-detection tasks (e.g., LSTMTF), and as originally proposed, with the predicted values fed back during testing (e.g., LSTMTFC). Furthermore, this chapter also introduced training and testing time measurements for the tested architectures.

The training and testing procedures were performed using a wide range of both internal and external hyperparameters, while the results were analyzed using various performance metrics. The models were tested in multiple configurations, namely multi-input single-output and multi-input multi-output using two prediction modes: many-to-many and many-to-one. For reproducibility, all the tested neural network configurations, hyperparameters, and datasets were documented throughout this chapter.

In both configurations, MISO and MIMO, the VLSTM (i.e., the standard LSTM without teacher forcing) obtained better results in terms of training convergence time for the M2M prediction method; however, in terms of prediction MAE, LSTMTF obtained better results. Out of the three neural networks, LSTMTFC obtained the worst results in terms of testing MAE using the M2M approach.

In every experimental scenario, there was a small decrease in the prediction error when switching from the MISO to the MIMO configuration on the same neural network type. However, the time taken for training and testing increased by 21% for training and 28% for testing when using the MIMO configuration. Overall, the input sequence length, mini-batch size, number of hidden units, and number of lags influenced the training and testing performance. Conversely, the learning rate's influence appeared to be smaller in all the experiments for all neural network architectures.

As illustrated by the experimental results, the architecture of LSTMs can be significantly reduced, while still maintaining prediction performance. This was observed while using fewer features for the MISO architecture, while still obtaining similar results to the MIMO architecture. Moreover, by using TF, it was shown that the models can be trained with a reduced number of samples while using only one hidden layer and still outperform other models. The reduced architecture (i.e., number of inputs, one hidden layer, and number of hidden units) and the possibility of training models with fewer samples make such models suitable candidates for real-time operations and on resource-constrained devices.

The proposed LSTMTF model was also compared with 11 additional parametric and nonparametric forecasting, regression, and prediction models, including some state-of-the-art ones. These additional experimental results illustrate the modeling capabilities of the proposed LSTMTF in terms of measured MAE values. In all experiments, the proposed model yielded lower MAE values compared to the 11 models tested.

Chapter 4

Feature Analysis for LSTMTF

This chapter enhances the LSTMTF model, as introduced in the previous chapter, by proposing two feature selection methodologies, an overfitting analysis approach, and a method for dealing with missing values in real-time. This chapter also presents extensive experimental evaluations and results for all proposed methods.

4.1 Feature Selection

In the direction of feature selection, this chapter introduces a correlation-based feature selection method. This method leverages the Pearson's coefficient score to select the group of inputs for each output variable. As part of the second approach, a new method inspired by sensitivity analysis [31] is introduced. This method includes feature ranking and automatic feature selection, for regression tasks. Although this method is proposed and tested on LSTMTF models, it is also suitable for other models. Following a backward approach, feature ranking is performed by sequentially eliminating inputs and measuring changes in model prediction residuals using the Energy Distance metric [32]. Automatic feature selection process includes a forward approach, starting with a predetermined number of features, and incorporates at each step additional features based on their ranks, until a stopping criterion is met.

4.1.1 Correlation Based Selection

This approach follows a correlation-based technique for selecting the groups of inputs and outputs. The selection process leverages Pearson's product-moment correlation coefficient [33]. Here, from numerous measured signals, the ones that exhibit a high correlation coefficient with the chosen output signal are selected. Pearson's product momentum correlation (Pearson's correlation) describes the strength of the relationship between variables.

4.1.2 Score Based Selection

In summary, the methodology for feature ranking utilizes a sensitivity analysis-based backward approach, which does not require retraining the LSTM models. The resulting list of ranked features is employed for feature selection. The selection process follows a forward-based approach where, based on a stopping criterion, the highest-ranking features are incrementally added one by one, and the model is retrained at each step. The same feature selection procedures are followed for multiple output variables. Furthermore, in both approaches, the Energy Distance is utilized to compute a distance score.

4.2 Overfitting Tests

To develop the overfitting tests, we begin with the premise that LSTM models with Teacher Forcing overfit the previous output ground truth value. Consequently, during inference, these models predict close to the previous ground truth value, ignoring the spatio-temporal relationship between the rest of the inputs and the output variable, this is *y-overfitting*.

If a model y-overfits, it introduces the following assumptions. First, disabling any of the additional inputs would not have any influence on the model's performance, as it "relies" only on the previous output ground truth value to make new predictions. Second, disabling all additional inputs would not affect the model's performance, based on the same assumption as above. Third, setting the output ground truth value to a constant would not affect the performance, as the model would only predict close to the previous ground truth value.

The approach outputs an overfitting score, measuring the distance between the prediction error distribution in various scenarios, using three distribution distance metrics.

4.3 Missing Values

For the LSTMTF model, it is assumed that the output ground truth values are available during both training and inference. Nevertheless, a significant issue is raised in this approach when the output ground truth values are missing due to unforeseen events (e.g., communication faults and erroneous sensor readings).

While many imputation techniques exist for dealing with missing data, very few of them are addressing dealing with missing data in real time scenarios. As a solution to this issue, this chapter introduces an approach that does not require using any additional models for imputation.

The proposed approach to deal with missing values involves switching the same model from using the output ground truth value, as an extra input, to using its previous predicted value, in real-time, when the target values are missing.

For this scenario, we are considering the case where only a small amount of values are missing, due to communication faults or due to erroneous measurements. In a real scenario, missing values over a prolonged period might be clear indicators of failures (prompting the initiation of internal alert mechanisms) and might even result in several systems shutting down, however, in the case where only a few measurements are missing, our proposed solution should continue functioning.

4.4 Summary

This chapter presented two approaches for feature selection utilizing Pearson's correlation coefficient and a novel score-based approach. The second proposed approach is inspired by sensitivity analysis and is applicable for regression tasks. In this direction, the Energy Distance was utilized for feature ranking, together with a forward-based feature selection methodology. The experimental results illustrated that the score-based approach can obtain notable and comparable results to well-established techniques, with low RMSE values and high R^2 values.

Additionally, this chapter proposed a novel method to test if models that utilize TF heavily rely only on the previous ground truth value and ignore other exogenous inputs (y-overfitting). In this case, these models might naively predict incorrect new values. This investigation focused specifically on the LSTMTF model and utilized three distribution distance metrics. The experimental evaluation results highlighted that all the selected distance metrics demonstrated that the model did not y-overfit in any of the tested scenarios.

This chapter also introduced a method for dealing with missing data in real time.

Chapter 5

Anomaly Detection using LSTMTF

This chapter explores two distinct directions from the field of anomaly detection in nonlinear systems, namely tampering and fault detection.

A significant number of tampering attempts have been observed in the automotive industry, specifically targeting environmental protection systems. These attempts have been identified in numerous studies such as [34] and [35]. Furthermore, as the severity of tampering reached critical levels, large-scale research and innovation projects have been dedicated to removing tampering in emissions-relevant systems, e.g., the Diagnostic Anti-Tampering Systems (DIAS) project [36].

Moving forward to the second applicability direction, namely fault detection. Fault detection can be seen as a specific application of anomaly detection focused on identifying deviations from the normal behavior of the system caused by sensor failures, component failures, or interventions in the system. Yoon and Macgregor in [26] state that "the purpose of fault detection is to determine the occurrence of an abnormal event in a process". Furthermore, as highlighted by Amini and Zhu [37] misclassifying normal samples as faults can result in unnecessary operational disruptions and increased labor costs.

5.1 Proposed Anomaly Detection Solutions

To address the previously described directions within the anomaly detection field, namely tampering and fault detection, this chapter introduces two anomaly detection ensembles. The ensembles leverage the modeling capabilities of LSTMTFs, which are used as a means to model the monitored nonlinear systems. The ensembles also incorporate Cumulative Sum (CUSUM) Control chart and Histogram distance-based detection approaches that monitor the changes in the LSMTF prediction residuals. Detector decisions are fused using two majority voting-based techniques.

During the detection phase, each detector monitors a signal (e.g., a variable) by analyzing the deviations from the normal learned behavior. Employing a threshold-based methodology, each detector outputs a binary decision regarding the validity of a new data point. The final decision is given by the ensemble using majority voting-based techniques.

5.1.1 Ensemble Architecture

A common element in the ensembles is the MISO predictive LSTMTF model. In both ensembles, the base detectors monitor one specific signal by constantly analyzing the prediction errors received from the predictor, using two distinct techniques. These detectors are the Cumulative Sum (CUSUM) Based Detector (CBD) and the Histogram Distance Based Detector (HBD). Employing a consistent terminology, the ensembles are labeled as CBE and HBE, based on the utilized detection method.

Individual decisions of the base detectors are combined using two methodologies. First, utilizing a majority voting scheme. Second, utilizing a novel Adaptive Majority Weighted Voting (AMWV) fusion methodology that takes into account the historical decisions of each detector and outputs one of three decisions: normal, alert, and warning.

5.1.2 Base Detectors Architecture

The CBD monitors changes, in both the mean and variance values of the LSTMTF prediction error, using two variants of the 1-CUSUM scheme [38]. The 1-CUSUM scheme has the ability to detect changes (i.e., increase and decrease shift) in both mean and variance values, using a single two-sided control chart, which works with single observations. The first proposed variant utilizes the CUSUM chart as originally proposed in [38], for a point-by-point CUSUM computation, while the second variant employs a sliding window methodology, which computes the CUSUM values over a sliding window.

The base detector of the second ensemble, named HBD, also utilizes LSTMTF predictive models but processes prediction errors differently. First, HBD constructs the histogram of the prediction errors over a given time window. Second, using a custom distance metric, it computes the distance between the histogram of the prediction errors of the training data and the histogram of the data contained in the current time window. Last, each detector outputs a binary decision using a threshold-based approach.

5.1.3 Adaptive Majority Weighted Voting Scheme

The proposed fusion technique, applied for tampering detection, is a modified version of the Majority Weighted Voting scheme, with an additional historical reputation component. That is, at each time step, the weights of the detectors are adjusted (e.g., increased or decreased) depending on whether the detector votes the same as the

majority or not. The weights are adjusted by a percentage, which is computed based on the previous decisions of said detector (e.g., if in the past, it has voted the same as the majority). This weight-adjustment methodology awards higher weights to detectors that vote the same as the majority compared to the detectors that more often disagree with the majority. Furthermore, the weights can drop to zero, thus temporarily *ignoring* that detector decision.

The final decision of the ensembles can be one of the following: normal, alert, or warning. The warning state is triggered when the majority vote does not trigger an alert but there is at least one detector that identified a tampered observation. Considering that each detector monitors a different signal, there is the possibility that tampering one or more signals might not affect the rest of signals, especially if the physical tampered component is not running in a closed loop. Thus, by generating a warning, further investigation can be carried out on that component.

5.2 Summary

This chapter first addressed a new emerging threat, namely tampering of the vehicle's environmental protection systems. Tampering can have serious effects on both human health and the environment, as tampered vehicles emit higher concentrations of pollutants, such as nitrogen oxides and particulate matter. Additionally, this chapter also introduced an ensemble-based approach to fault detection in continuous nonlinear systems.

In response to the former threats, this chapter proposed two ensemble-based methodologies for tampering detection. The proposed solutions utilize predictive LSTMTF models in conjunction with CUSUM and histogram distance-based detectors. The CUSUM and histogram distance-based detectors receive as input the prediction error from the predictive models and output a binary decision using a threshold-based approach.

When applied to tampering detection, the proposed solutions obtained notable results, including 0% False Positive Rates on all datasets and up to 100% detection rates in most cases. Furthermore, the ensembles were compared to state-of-the-art tampering detection methodologies with promising results. This chapter also provides resource consumption and scalability measurements on a reference embedded system, demonstrating the possibility of integrating the proposed solutions in an actual embedded environment.

In the context of fault detection, the developed ensemble technique demonstrated notable efficiency by achieving a 100% detection rate with 0% FPR for every unknown fault. The same results were observed even on challenging faults, including faults 3, 9, and 15. Furthermore, the ensemble technique outperformed numerous other detection methods from the scientific literature.

Chapter 6

Conclusions

In alignment with the first and fifth research objectives, the third chapter introduced an approach for nonlinear system modeling, namely the LSTMTF model. This model encompasses an LSTM model with a modified version of the Teacher Forcing algorithm applied during both training and inference for datasets originating from continuous nonlinear systems.

To validate the LSTMTF model, an extensive hyperparameter and benchmark analysis was performed on a popular reference dataset. This extensive evaluation included various hyperparameters and multiple model architectures. Furthermore, the prediction performance of the LSTMTF model was compared with 15 state-of-the-art modeling techniques, with promising results. The experimental results also revealed that the LSTMTF model can model the behavior of nonlinear systems even with reduced architectures and complexity.

In alignment with the second and fifth research objectives, the fourth chapter introduced two feature selection methodologies. The first feature selection approach uses Pearson's correlation coefficient between possible input candidates and the output variable during the selection process. The second proposed approach encompasses a novel feature selection and ranking methodology. This approach utilizes the Energy Distance first as a means to quantify the distances between the prediction residuals and compute a score, which is later utilized in the ranking and selection procedures. The latter feature selection approach was experimentally compared with 7 state-of-the-art regression feature selection approaches.

As stated above, the LSTMTF model utilizes the TF algorithm during both training and inference, where the previous observed output value is fed as an additional input at each time step. This raised the question whether models that utilize this TF approach heavily rely only on the previous ground-truth value and ignore other exogenous inputs. In this direction, three assumptions were introduced and empirically tested, utilizing a methodology inspired by sensitivity analysis. To analyze these assumptions, different

scenarios were generated by sequentially disabling inputs or setting constant ground truth values for the output variable.

In addition, the fourth chapter also introduced a method for dealing with missing data in real time. The experimental results highlighted that using this method, the model can effectively handle missing values while making predictions, improving the robustness and reliability of the LSTMTF model.

To address the last three research objectives, in the fifth chapter, the LSTMTF model was employed as part of two novel proposed anomaly detection ensembles. These ensembles also encompass CUSUM and Histogram distance-based detectors which are utilized to monitor the LSTMTFs prediction residuals. The CUSUM-based detectors function in two modes, point-by-point and window-based. For the second type of detector, which monitors the changes in the distribution of the residuals, a new distance metric was proposed and formally demonstrated. The detector decisions are fused using two approaches. First, utilizing a majority voting scheme. Second, using a novel adaptive weighted majority voting scheme that accounts for the historical decision of each detector in the weight adjustment procedures.

The proposed anomaly detection ensembles were validated in two distinct scenarios from the field of anomaly detection, namely automotive tampering and nonlinear system fault detection. The experimental results performed on four datasets highlighted the validity of the proposed ensembles, revealing low false alerts and high detection rates in most of the tested tampering scenarios and analyzed faults. In both anomaly detection scenarios, the proposed ensembles were compared with 24 state-of-the-art proposed solutions, with promising results. Specifically, the proposed ensembles outperformed other methods in terms of false positive rates, true positive rates, and detection delays in most experimental scenarios.

The fifth chapter also presented resource measurement results, performed on an embedded device with limited resource capabilities. These measurements were performed to verify the possibility of integrating our approaches in resource-constrained embedded devices and included CPU, Memory, and Scalability measures. The results of these experiments demonstrated the possibility of integrating the proposed solutions in a real embedded environment with low resource consumption impact.

6.1 Scientific Contributions

This section summarizes the significant scientific contributions of the thesis.

- **Chapter 2:**
  1. Extensive review of the scientific literature in the direction of the thesis. This includes nonlinear systems, machine learning-based modeling, neural network architectures, hyperparameter selection, feature analysis, and two domains of application for anomaly detection.

- **Chapter 3:**
  1. The design and development of the LSTMTF model, a Long Short-Term Memory model with a variant of Teacher Forcing, for nonlinear system modeling utilizing time series data.
  2. Extensive benchmark and hyperparameter analysis of the LSTMTF model on a popular dataset originating from a representation of a reference nonlinear system. This evaluation includes various hyperparameters (e.g., sequence length, mini-batch size, learning rate, hidden units, TF lags) and multiple model architectures and prediction modes (e.g., Multi-Input Single Output, Multi-Input Multi Output, Many-to-Many, Many-to-One).
  3. Training/Testing time measurements for the analyzed LSTMTF architectures.
  4. Extensive comparisons with 15 state-of-the-art approaches used in the scientific literature for nonlinear modeling. This includes well-established and state-of-the-art parametric and non-parametric forecasting, regression, and prediction models.
  5. Performance comparison of the LSTMTF model and other prediction algorithms with manual selection and automatic hyperparameter optimization.
- **Chapter 4:**
  1. The design and development of a novel feature analysis methodology for regression models that encompasses backward-based feature ranking and forward-based feature selection techniques.
  2. Performance comparisons with 7 well-established feature selection techniques.
  3. Complexity and training/testing time measurements for the LSTMTF model with and without feature selection.
  4. An in-depth overfitting analysis of the LSTMTF model, utilizing three distribution distance metrics and testing three proposed overfitting assumptions.
  5. The design of a methodology to deal with missing values during testing utilizing the LSTMTF model. Missing values caused by erroneous sensor reading or communication errors.
- **Chapter 5:**
  1. The design and development of an ensemble-based anomaly detection framework that encompasses the following components:
     - The design of Cumulative Sum and Histogram Distance-based anomaly detectors capable of two operation modes, namely point-by-point and window-based.
     - The design and development of unsupervised anomaly detection ensembles that incorporate LSTMTF predictors together with the detectors designed above.

   - The design of a new adaptive majority-weighted voting fusion scheme for the ensemble. This fusion scheme accounts for the historical decisions of the detectors in the weight adjustment procedures.
2. The design and development of new parameter selection approaches for the anomaly detection ensembles.
3. The applicability of the anomaly detection ensembles to solve real-world issues, namely automotive tampering and nonlinear system fault detection, with promising results.
4. In both anomaly detection directions, the ensembles are compared with a total of 24 related anomaly detection approaches, with promising results in terms of high detection rates, low false alerts, and detection delays. Additionally, in the fault detection direction: the successful detection with high accuracy of three faults that are considered difficult to detect by numerous researchers throughout the literature.
5. Performance measures in terms of CPU, memory usage, and scalability on resource-limited devices.

6.2 Future Work

As shown in the fifth chapter, supervised techniques that utilize classifiers, trained with both anomalous and clean observations, yield superior results in comparison. However, the scarcity and sparsity of available datasets still demand additional research towards unsupervised approaches. An interesting future research direction involves utilizing unsupervised detection approaches capable of identifying and classifying the source of the anomaly (e.g., anomalous signal). Moreover, the integration of explainable artificial intelligence techniques remains an open and interesting research direction.

An additional possible direction is the exploration of various other unsupervised detection approaches and techniques, apart from neural networks. Future research could focus on developing approaches that are trained with limited anomaly-free observations but are capable of detecting abnormal behaviors with high accuracy. Specific research directions might include the introduction of novel techniques and algorithms that can adapt to evolving system behaviors.

Although feature analysis, together with missing data techniques, remain widely researched directions, further study in various other domains, on datasets originating from both linear and nonlinear systems, might provide superior solutions.

References

[1] Johan Schoukens and Lennart Ljung. "Nonlinear system identification: A user-oriented road map". In: *IEEE Control Systems Magazine* 39.6 (2019), pp. 28–99.

[2] Bilash Kanti Bala, Fatimah Mohamed Arshad, Kusairi Mohd Noh, et al. "System dynamics". In: *Modelling and Simulation* 274 (2017).

[3] José Maria P Menezes Jr and Guilherme A Barreto. "Long-term time series prediction with the NARX network: An empirical evaluation". In: *Neurocomputing* 71.16-18 (2008), pp. 3335–3343.

[4] Olalekan Ogunmolu et al. "Nonlinear systems identification using deep dynamic neural networks". In: *arXiv preprint arXiv:1610.01439* (2016).

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[6] Czako Zoltan and Sebestyen-Pal Gheorghe. "Self-Adaptive Artificial Intelligence Techniques Used for Anomaly Detection". In: *PhD Thesis* (2023).

[7] Thomas G Dietterich. "Ensemble methods in machine learning". In: *International workshop on multiple classifier systems*. Springer. 2000, pp. 1–15.

[8] Xuezheng Jiang et al. "An adaptive multi-class imbalanced classification framework based on ensemble methods and deep network". In: *Neural Computing and Applications* 35.15 (2023), pp. 11141–11159.

[9] Kishan G Mehrotra et al. "Ensemble Methods". In: *Anomaly Detection Principles and Algorithms* (2017), pp. 135–152.

[10] Charu C Aggarwal et al. *Outlier detection for temporal data*. Morgan & Claypool Publishers., 2014.

[11] Andrew A Cook, Göksel Mısırlı, and Zhong Fan. "Anomaly detection for IoT time-series data: A survey". In: *IEEE Internet of Things Journal* 7.7 (2019), pp. 6481–6494.

[12] Pang Clement. *Anomaly Detection on Time Series Data*. https://patents.justia.com/patent/20200034733. U.S. Patent 20200034733. 2020.

[13] Ane Blázquez-García et al. "A review on outlier/anomaly detection in time series data". In: *ACM Computing Surveys (CSUR)* 54.3 (2021), pp. 1–33.

[14] Paul-Adrian Călburean et al. "Prediction of 3-year all-cause and cardiovascular cause mortality in a prospective percutaneous coronary intervention registry: Machine learning model outperforms conventional clinical risk scores". In: *Atherosclerosis* 350 (2022), pp. 33–40. ISSN: 0021-9150.

[15] Thyago P Carvalho et al. "A systematic literature review of machine learning methods applied to predictive maintenance". In: *Computers & Industrial Engineering* 137 (2019), p. 106024.

[16] Sanda-Maria Avram and Mihai Oltean. “A Comparison of Several AI Techniques for Authorship Attribution on Romanian Texts”. In: *Mathematics* 10.23 (2022). ISSN: 2227-7390.

[17] Ali Bou Nassif et al. “Machine learning for anomaly detection: A systematic review”. In: *Ieee Access* 9 (2021), pp. 78658–78700.

[18] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[19] Lennart Ljung et al. “Deep learning and system identification”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 1175–1181.

[20] Furkan Elmaz and Özgün Yücel. “Data-driven identification and model predictive control of biomass gasification process for maximum energy production”. In: *Energy* 195 (2020), p. 117037. ISSN: 0360-5442.

[21] Ronald J Williams and David Zipser. “A learning algorithm for continually running fully recurrent neural networks”. In: *Neural computation* 1.2 (1989), pp. 270–280.

[22] Alex Sherstinsky. “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network”. In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306.

[23] Thomas M Breuel. “Benchmarking of LSTM networks”. In: *arXiv preprint arXiv:1508.02774* (2015).

[24] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. “The performance of LSTM and BiLSTM in forecasting time series”. In: *2019 IEEE International Conference on Big Data (Big Data)*. IEEE. 2019, pp. 3285–3292.

[25] Kishan G Mehrotra et al. *Anomaly detection*. Springer, 2017.

[26] Seongkyu Yoon and John F MacGregor. “Fault diagnosis with multivariate statistical models part I: using steady state fault signatures”. In: *Journal of process control* 11.4 (2001), pp. 387–400.

[27] James J Downs and Ernest F Vogel. “A plant-wide industrial process control problem”. In: *Computers & chemical engineering* 17.3 (1993), pp. 245–255.

[28] Cory A. Rieth et al. “Additional Tennessee Eastman Process Simulation Data for Anomaly Detection Evaluation”. In: *Harvard Dataverse*. Harvard Dataverse, 2017.

[29] Florian Schmidt. “Generalization in generation: A closer look at exposure bias”. In: *arXiv preprint arXiv:1910.00292* (2019).

[30] Boris Hanin. “Which Neural Net Architectures Give Rise to Exploding and Vanishing Gradients?” In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[31] Daniel S Yeung et al. *Sensitivity analysis for neural networks*. Springer, 2010.

[32] Maria L Rizzo and Gábor J Székely. “Energy distance”. In: *wiley interdisciplinary reviews: Computational statistics* 8.1 (2016), pp. 27–38.

[33] Jean Dickinson Gibbons and Subhabrata Chakraborti. *Nonparametric statistical inference*. CRC press, 2014.

[34] Barouch Giechaskiel et al. “Effect of Tampering on On-Road and Off-Road Diesel Vehicle Emissions”. In: *Sustainability* 14.10 (2022). ISSN: 2071-1050.

[35] CALEB Braun et al. “Heavy-Duty Emissions Control Tampering in Canada”. In: *International Council Clean Transportation (ICCT) Report* (2022).

[36] DIAS. *Diagnostic Anti-Tampering Systems*. https://Dias-Project.Com. Accessed: 2024-01-05.

[37] Nima Amini and Qinqin Zhu. “Fault detection and diagnosis with a novel source-aware autoencoder and deep residual neural network”. In: *Neurocomputing* 488 (2022), pp. 618–633.

[38] Zhang Wu and Qinan Wang. “A single CUSUM chart using a single observation to monitor a variable”. In: *International Journal of Production Research* 45.3 (2007), pp. 719–741.