Detecting Anomalies in Industrial Control Systems with LSTM Neural Networks and UEBA

Camilo Piñón-Blanco
GRADIANT
Vigo, Spain
cpinon@gradiant.org

Fabián Otero-Vázquez
GRADIANT
Vigo, Spain
fovazquez@gradiant.org

Ines Ortega-Fernandez
GRADIANT
Vigo, Spain
iortega@gradiant.org

Universidade de Vigo
Vigo, Spain
ines.ortega@uvigo.es

Marta Sestelo
Universidade de Vigo
Vigo, Spain
sestelo@uvigo.es

***Abstract*—The increasing adoption of the Industrial Internet of Things and integration of operational technology with information technology networks have made industrial control systems (ICS) more vulnerable to cyber-attacks, which can cause severe consequences such as disruption of critical infrastructure, loss of data, and significant financial losses. To enhance the security and resilience of these systems, anomaly detection in ICS has gained significant attention in recent years. This paper introduces ongoing research focused on using Long Short-Term Memory (LSTM) neural networks for forecasting and subsequent anomaly detection over device logs. This approach involves User and Entity Behaviour Analytics (UEBA) to analyze and define entities of interest from a real industrial plant and extract a baseline behaviour model through features that are fed into the LSTM model for predicting future events and detecting anomalies. The proposed solution has the potential to provide real-time detection of cyber and physical threats, thereby enhancing the security and resilience of industrial control systems.**

***Index Terms*—Anomaly Detection, Industrial Control Systems, User and Entity Behaviour Analytics, Neural Networks, Long-Short Term Memory**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

In the past decade, there has been a significant increase in security and safety incidents in industrial environments and critical infrastructure. Some of these incidents have led to devastating consequences, such as the Stuxnet (2010) [1] computer worm's takeover of several Programmable Logic Controller (PLCs), resulting in the destruction of centrifuge tubes at a uranium enrichment plant in Iran. Malware attacks like BlackEnergy (2015) [2] and Industroyer (2016) [3] on Ukrainian power grids caused outages that affected thousands of users. These events highlight the vulnerability of critical infrastructures to cyber-attacks, as well as the need for effective means of detecting spurious behaviour that may represent the first signs of a threat.

Modern industrial control systems incorporate both Operative Technology (OT) and Information Technology (IT) elements, generating massive amounts of data that require processing to extract valuable insights [4]. A considerable fraction of this data comes from measurement sensor logs distributed throughout the production chain in the form of multivariate time series. To overcome the limitations of traditional methods, like Statistical Process Control techniques such as CUSUM or EWMA [5], researchers have increasingly turned to machine learning (ML) based anomaly detection [6] and (UEBA) [7] for system monitoring and anomaly detection.

UEBA is a learning paradigm where machine learning, statistical analysis, and data aggregation are integrated to detect trends and patterns accurately [8]. A significant advantage of UEBA tools is their ability to aggregate knowledge about the behaviour patterns of individuals and to detect anomalous patterns that may indicate potential security threats [9]. UEBA is a key feature of state-of-the-art Security Information and Event Management systems for identifying user, device, and application behaviours [10]. Firstly, baseline behaviour profiling is achieved by modelling various data sources as feature vectors representing attributes and characteristics of devices. These vectors are then leveraged to train a machine learning model that can learn from the historical actions of devices and machinery. Potential threats are therefore detected whenever baseline behaviours are violated.

The present work studies the use of a Long Short-Term Memory (LSTM) architecture, a deep learning (DL) method based on an artificial Recurrent Neural Network (RNN), to forecast future sensor measurements aggregated for each entity present in an industrial facility (e.g. liquid and gas tanks, pumps, valves, etc). The model is trained on historical multivariate data and the predicted values are then compared to the actual measurements. To identify anomalies, we propose the use of a decision threshold automatically computed over residual error obtained in the validation set. This threshold allows us to detect deviations from the expected values and determine whether they represent an abnormal behaviour of the specific entity that is being modelled). By combining the LSTM model with the decision threshold, we aim to provide an accurate and reliable method for predicting sensor values and identifying potential anomalies in real-time.

The proposed approach is based on the underlying assumption that both active cyber and physical threats tend to have a discernible impact on the normal functioning of the system, which may be detected by monitoring certain system variables [11]. For example, in the chemical industry, a cyber-attack may cause a sudden change in the flow rate of raw materials or a physical attack may damage critical components, leading

to abnormal readings in temperature or pressure sensors. By analyzing these variables, it is possible to detect deviations before their impact on the system escalates. This information can then be used to take appropriate actions to mitigate the impact of the threat and prevent further damage to the system.

LSTM networks have been widely studied in the literature for time series analysis in multiple applications, including forecasting and anomaly detection in ICSs [12]–[15]. However, the present work aims to explore the use of these models enabled by UEBA to provide contextual knowledge about the plant design and the baseline behaviour of different entities, allowing the combination of information from multiple sources at the feature level.

The remainder of this paper is structured as follows: Section II presents a brief review of the relevant literature, discussing the state-of-the-art techniques related to anomaly detection in ICSs. Section III describes the research design and the main elements of our proposed methodology. Section IV presents the current findings of the study, providing a detailed discussion of the results. Section V summarises the key findings of the ongoing research and outlines the directions and next steps for future research, reflecting the limitations of the current stage of the study and providing recommendations for addressing them.

II. Related work

Several methodologies have been proposed to address threat and anomaly detection in industrial environments, and several studies have investigated their effectiveness [16]. These are typically grouped into two categories: signature-based detection and behaviour analytics. The former is based on a knowledge base of known threats to compare signatures with the current state of the network and its devices, whereas the latter focuses on comparing user and/or entity behaviour historically to determine a baseline and detect anomalies against it. This provides a more flexible approach, allowing the detection of zero-day attacks that have not been observed previously. In this section, we briefly review recently published work in the field of behavioural analytics for anomaly detection on ICSs, focusing on the different data sources used to model normal industrial behaviour and the corresponding techniques applied to each one.

One of the main areas of research focuses on the use of data generated due to network activity. In *Ortega et al.* [17], the authors study the use of flow (NetFlow) features to build a network (agnostic) intrusion detection system based on deep autoencoders, showing their ability to outperform traditional machine learning methods like Isolation Forest [18]. On the other hand, *Liu et al.* [19] introduce a deep learning framework that uses LSTM, Convolutional Neural Networks (CNN), and Multi-head Self Attention Mechanism to detect anomalies in packet payloads. This method achieves a high detection rate and a low false positive rate but comes at the expense of the inherent computational and memory requirements of deep packet inspection.

Process mining techniques have also been the focus of researchers due to their ability to model operational processes based on event data. *Myers et al.* [20] introduce conformance checking over ICS data logs to detect unusual behaviour and cyberattacks. These logs, especially those related to sensors and actuators' activity have also been extensively used as inputs to various ML and DL models. In *Elnour et al.* [21], a Dual-Isolation-Forests-Based framework is presented, modelling the device data as a tabular input and obtaining state-of-the-art detection performance. Alternatively, *Li et al.* [22] model the sensor measurements as a multivariate time series on which a Generative Adversarial Network (GAN) is trained to identify deviations from the normal behaviour.

With regard to the use of UEBA, recent advancements have demonstrated its effectiveness in improving security systems. *Martín et al.* [23] present the combination of user behavioural information at the feature level to enhance the performance of continuous authentication systems.However, most of the work focuses on enterprise or information technology environments, while the use of UEBA on industrial settings is not widely studied. *Nocera et al.* [24] present a UEBA-based solution to mitigate Distributed Denial of Service (DDoS) attacks in a Cloud computing environment. They model the behaviour of users through an LSTM neural network in order to detect potential bots (illicit users) in the system. *Shashanka et al.* [25] propose a UEBA module for enterprise security that tracks and monitors behaviours of users, IP addresses and devices in a company, establishing a baseline for anomaly detection through Singular Value Decomposition (SVD) based techniques. *F. Rashid and A. Miri* [26] have recently shown the feasibility of UEBA approaches even in differentially private data contexts, where information disclosure to third parties is a critical matter, making it necessary to modify the original data by means of noise insertion while keeping the analysis and detection capabilities.

III. Methodology

This section presents the key components of the proposed methodology that enable the modelling of the normal behaviour of various entities in an industrial plant. Firstly, we outline the utilization of User and Entity Behavior Analytics (UEBA) to construct robust entity models that improve the anomaly detection process. Secondly, we discuss Long Short-Term Memory (LSTM) networks, highlighting their primary constituents and their applicability in anomaly detection. Lastly, we present the architecture for data collection, preprocessing, and model training.

A. User and Entity Behaviour Analytics

Industrial facilities have numerous devices (which will be the entities of our UEBA model). By leveraging knowledge about the mapping between extracted variables from system monitoring and the different entities, we will be able to characterize them through multivariate modelling. The objective is to have one model per entity. In this way, a compromise middle ground is reached between modelling each of the variables individually (which is computationally expensive and does not allow modeling the relationships between variables) and modelling them all together, which may introduce biases in case they are variables associated with unrelated processes.

Entities and their corresponding features are mapped in a table, which is used to prepare the feature vectors by aggregating data points that correspond to those entities. This implementation allows customizing how the features will be treated and defining different model characteristics for

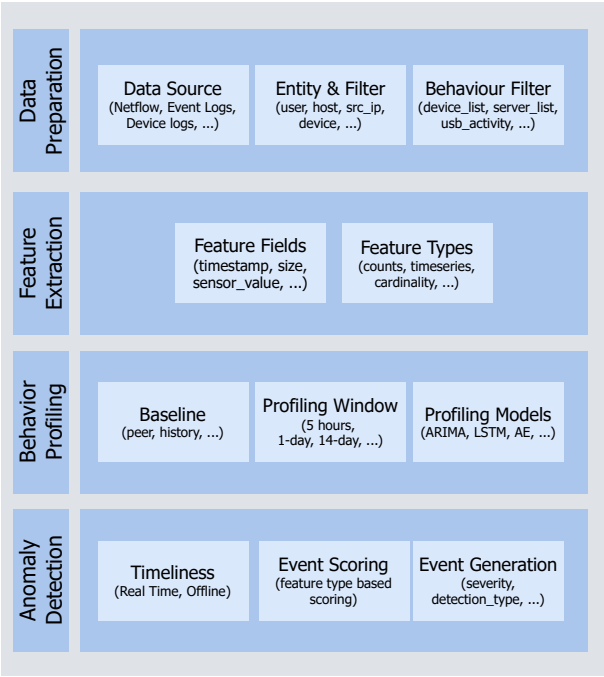

Fig. 1. Main elements of a UEBA architecture

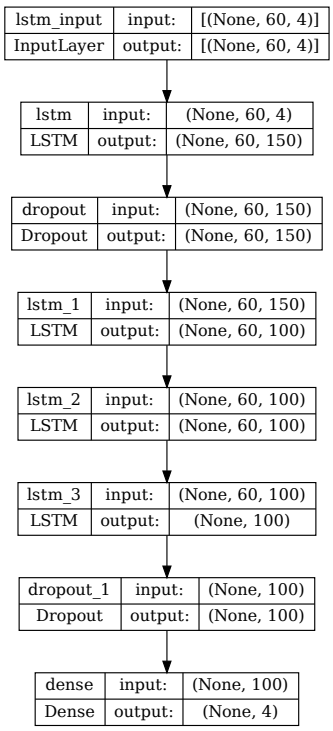

Fig. 2. Proposed LSTM model architecture

different data types. Once the feature vectors are prepared, the UEBA model uses a self-supervised learning algorithm to learn the baseline behaviour of each device. Since the UEBA model maintains one separate machine-learning model for each entity, it can detect anomalies specific to that entity, rather than detecting general anomalies across the entire system. This allows for faster detection and more targeted remediation of anomalies. Additionally, the UEBA model can adapt to changes in the system by continuously updating the models based on new data. This enables the model to remain effective over time as the system evolves.

Overall, the UEBA approach provides a comprehensive and flexible approach to anomaly detection in industrial systems. By using multivariate modelling and mapping entities to their associated variables, the model can accurately capture the complex relationships between variables and entities. This allows for more accurate anomaly detection and faster remediation.

Fig. 1 illustrates the main elements of a generic UEBA architecture [27] that has been considered for the design of our solution.

B. Long Short Term Memory networks

LSTMs are a type of Recurrent Neural Network designed for sequence analysis and prediction [28]. Their main feature is the use of a memory cell capable of storing information for long periods of time, as well as forgetting it if it is not relevant to the model anymore. This structure allows LSTMs to overcome the vanishing gradient problem [29], a common issue with neural networks trained using backpropagation.

The format of the input data in the recurrent networks is three-dimensional and is defined by three parameters: *batch size*, which specifies the number of observations that an LSTM model can process at a time, *timesteps*, which is the size of the input sequences and *input_dim* dimension of the sample, in this case, the number of sensor measurements. Fig. 2 depicts the details of the architecture which consists of a sequence of stacked LSTM layers combined with *dropout* layers, a regularization technique that randomly drops out (sets to zero) some of the inputs to the layer in order to prevent overfitting [30]. The activation function for the LSTM layers is the hyperbolic tangent *tanh*. It offers fast converging times and has a derivative that is computationally efficient to compute, which is important for training the network using backpropagation. The output layer is a *dense* layer with linear activation since values are unbounded. This layer has the same number of output features as the number of variables that model the entity, which produces the multivariate output of the model. The implementation has been carried out using the Keras Sequential API for Python [31]. The model is compiled using the Adam optimizer [32] and the Huber loss function [33], which is a robust loss function that is less sensitive to outliers than the mean squared error loss.

C. System Architecture

This section describes the main stages of the proposed architecture, summarised in Fig. 3. The proposed architecture consists of three stages: data preprocessing, model training and validation for threshold computation, and anomaly detection. During data preprocessing, input data is aggregated by entity, normalised, and transformed into a self-supervised learning setting. The normalised data is used for model training and inference. During model training, an independent LSTM network is trained to minimise the Huber loss for each entity. A validation set is used to compute a decision threshold for anomaly detection based on the 95th quantile

of the root mean squared error (RMSE). At last, during the anomaly detection stage, new values are predicted, and if the RMSE is greater than the decision threshold, the observations are flagged as an anomaly.

1) Data preprocessing: The preprocessing pipeline consists of three steps. First, the feature grouping by entities introduced in Section III-A is carried out, and a subset of data containing only the corresponding features will be fed to every entity model. Then, data normalisation is performed so that every feature is in the $[0, 1]$ range, which helps to stabilize the gradient descent step and helps models converge faster for a given learning rate. Finally, the data is transformed into a self-supervised learning setting. The time series is converted into a collection of input and output pairs of observations, where each input observation is a sequence of values over a specified number of time steps (lags) and each output observation is a single data point corresponding to the next time step (target value). This transformation enables a type of modeling called auto-regression. In auto-regression, a model is built using the past recent values (lags) of a time series as explanatory variables to predict future observations. We perform one-step ahead forecasting, where only the next time step is predicted for a given input sequence. From our initial validation, we have found that a fixed window of 60 time steps (5 hours) works best against higher or lower sizes. This lag window refers to the number of past time steps that are considered by the model when making predictions. In this case the model considers the 60 most recent time steps when predicting the next value in the time series. At this point, we split the training dataset into 80% for training and 10% for validation and threshold calculation, and 10% for model evaluation. This last model evaluation data will be used as a test set in our experiments to compute performance metrics in Section IV. For a real-time setting, all available historical data would be used for training and validation exclusively.

2) Offline Model Training and Validation using historic data: During the training phase, the preprocessed input sequences are fed into the neural network. To find the difference between the input sequence and output sequence, the Huber loss is computed. The loss is minimized during the training and the weights are updated during backpropagation. The model is trained over a set of normal observations, allowing the model to learn the normal behaviour during the training phase. Since the model is trained in normal data only, the model is expected to produce a higher error on anomalies, allowing us to detect them by establishing a decision threshold.

After the model is trained, its effectiveness is measured on the validation set. Furthermore, this stage computes the decision threshold that will be used for anomaly detection. The decision threshold is calculated from the row-wise Root Mean Squared Error (RMSE) for the predictions obtained from the validation sample. Then, the 95^{th} quantile of the RMSE is set as the decision threshold for anomaly detection. It is important to note that this quantile can be modified to regulate the sensitivity of the system to false positives.

3) Real-Time Forecasting and Anomaly Detection: New values in the time series are inferred using the trained LSTM model, taking into account the previous 60 time steps, referred to as the lag window. Then, the real value is compared to the predicted value. If RMSE exceeds the decision threshold, it gets flagged as an anomaly. Since the model is trained to perform an accurate prediction, when it detects a never-seen window (due to potential anomalies), it will produce poor predictions, leading to a higher error than the decision threshold, resulting in a flagged anomaly.

IV. Preliminary results

As we have seen, the architecture comprises Real-Time Forecasting and Anomaly Detection modules to determine whether an observation is an anomaly. Therefore, the evaluation of our approach consists of two aspects: assessing the forecasting capabilities of the model and evaluating the threshold-based anomaly detection methodology over the test set.

To carry out an initial validation of our research, the methodology is evaluated against a dataset collected from real data from 3 years of activity from a chemical plant. The dataset is composed of 12 features, 9 of which represent measurements (mainly temperatures and pressures) from sensors distributed across different elements of the infrastructure of the plant. The remainder three features are the timestamps and codes related to the product that is being manufactured. For this study, we choose to model the normal behaviour of a chemical processing tank. Therefore, this tank is our entity of interest, and we are modelling it by filtering 4 features (out of the 9 mentioned above) that correspond specifically to the operation of a tank: two of them related to temperature and two related to the pressure levels inside the tank.

The dataset has 294542 observations, obtained by recording the sensors' measurements with a sampling period of 5 minutes. We have trained the LSTM model with 235634 observations (80% of the dataset). The validation and threshold computation was carried out over 29514 observations (10% of the dataset) and the remaining 10% is used for testing.

Firstly, it is necessary to verify the model's ability to predict future values. To do so, we will make use of common evaluation metrics for time series forecasting, mainly focusing on the RMSE, which has the benefit of penalizing large errors more than other measures like the mean absolute error. Given a $m-$dimensional vector of time series variables $\mathbf{y}_i$, for $i = 1, \ldots, n$, the RSME is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \hat{\mathbf{y}}_i)^2}$$

being $\hat{\mathbf{y}}_i$ the predicted value by the model for the observation $\mathbf{y}_i$.

In order to facilitate the interpretation and assessment of the obtained RMSE values, we propose using the Normalised RMSE (NRMSE). Normalizing the RMSE allows the comparison between features with different scales. We will use one of the normalization approaches proposed in [34], which can be expressed as:

$$NRMSE = \frac{RMSE}{\mathbf{y}_{max} - \mathbf{y}_{min}},$$

where y_{max} and y_{min} are the maximum and minimum values of the series over the entire test interval (time horizon).

Fig. 4 shows the predicted and original time series of each sensor. We can observe how the proposed model is able to

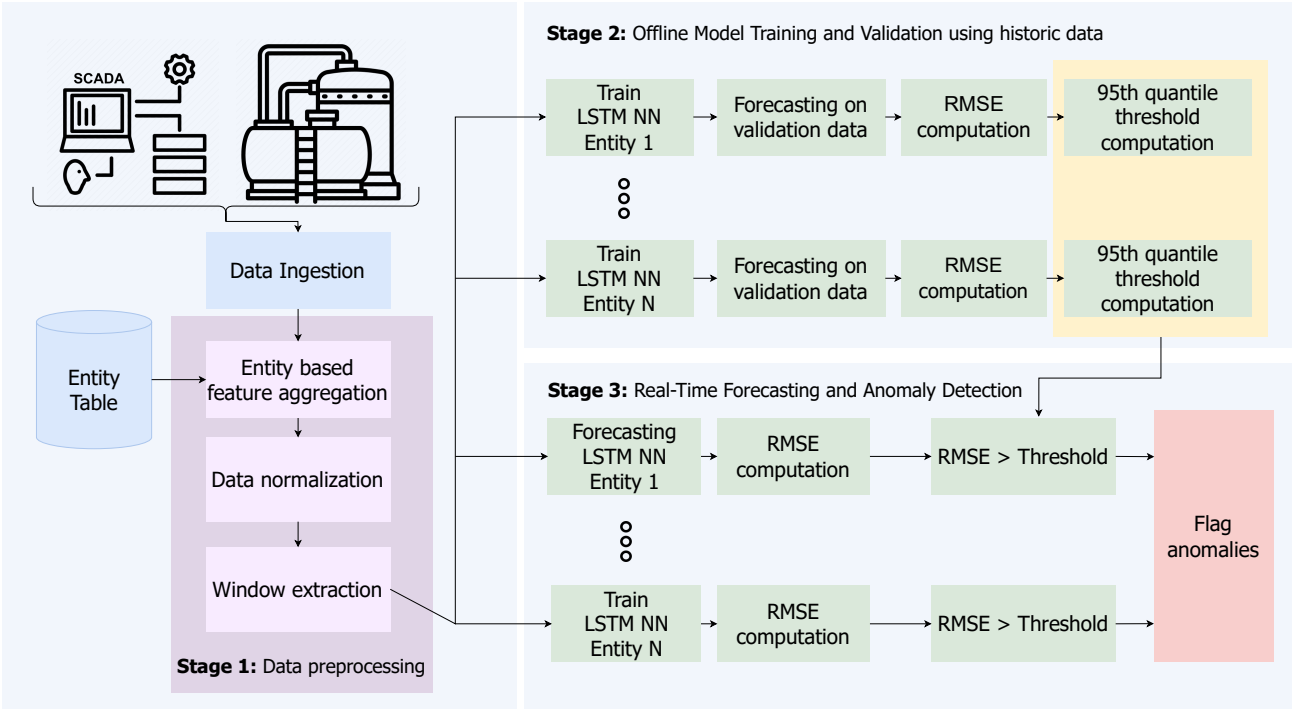

Fig. 3. System architecture

properly learn the trend of the original time series, with the exception of some large peaks, which is a common issue with LSTM forecasting models [35] [36]. Table I summarises the obtained RMSE for each sensor: overall, the algorithm performed better for Temperature Sensor 1 compared to Temperature Sensor 2 and for Pressure Sensor 2 compared to Pressure Sensor 1. If we consider the NRMSE, all sensor measurement predictions lay between 4.32 and 8.14 per cent. In Section V we highlight the need to compare these results with state-of-the-art approaches in order to fully validate our solution.

TABLE I
RMSE AND NRMSE FOR DIFFERENT SENSOR MEASUREMENTS FORECASTING VALUES

	Temperature Sensor 1 (℃)	Temperature Sensor 2 (℃)	Pressure Sensor 1 (millibar)	Pressure Sensor 2 (millibar)
RMSE	7.552	11.080	47.921	47.550
NRMSE	0.0460	0.0814	0.0435	0.0432

Since the data was collected from a real environment, there are no actual labels that indicate whether an observation is anomalous or not. However, we have no evidence that there has been a significant anomaly during the operation during the 3 years of data capture. Therefore, in order to validate the proposed anomaly detection framework, we generate synthetic anomalies in the data: we define a function that modifies some position y_{ij} of the $m-$dimensional vector of time series by inserting anomalies. The function first calculates the mean and the standard deviation of each dimension of the vector of time series. It then randomly selects a subset of the observations to insert anomalies. For each selected y_{ij}, it generates a new observation y^*_{ij} by adjusting the mean of the variable by a factor according to an anomaly intensity parameter (λ). The adjustment can be positive or negative, randomly chosen. λ acts as a scaling factor for the standard deviation of the selected feature, where a higher value of λ implies that the anomalies will be further away from the mean, making them more detectable by the anomaly detection framework. Therefore, the larger the value of λ, the greater the impact of the anomaly on the data, which increases the likelihood of detecting the anomaly.

The following expression represents how anomalies are inserted according to the intensity parameter λ:

$$y^*_{ij} = \hat{\mu}_j + \lambda \hat{\sigma}_j,$$

where $\hat{\mu}_j$ is the mean obtained from the j dimension of the vector of time series and $\hat{\sigma}_j$ is its standard deviation, for $j = 1, \ldots, m$.

Using the procedure described above, 5% of the dataset has been modified to insert synthetic anomalies in accordance with the 95$^{\text{th}}$ quantile used in the threshold calculation. These anomalies have been labelled and used for the calculation of metrics commonly used in binary classification problems and anomaly detection in unbalanced datasets.

There are four types of basic evaluation metrics for a classification model. The model is considered to perform well if it correctly identifies anomalies (True Positive, TP) and normal operation (True Negatives, TN). However, if the model incorrectly identifies an anomaly as normal (False Negative, FN) or a normal observation as an anomaly (False Positive, FP), this is considered an erroneous classification. For binary anomaly detection problems, there are several metrics derived from the ones mentioned above that can be used to evaluate the performance of a model, including:

- True Positive Rate (TPR), also known as sensitivity, measures the proportion of actual anomalous data points that are correctly classified as anomalous.

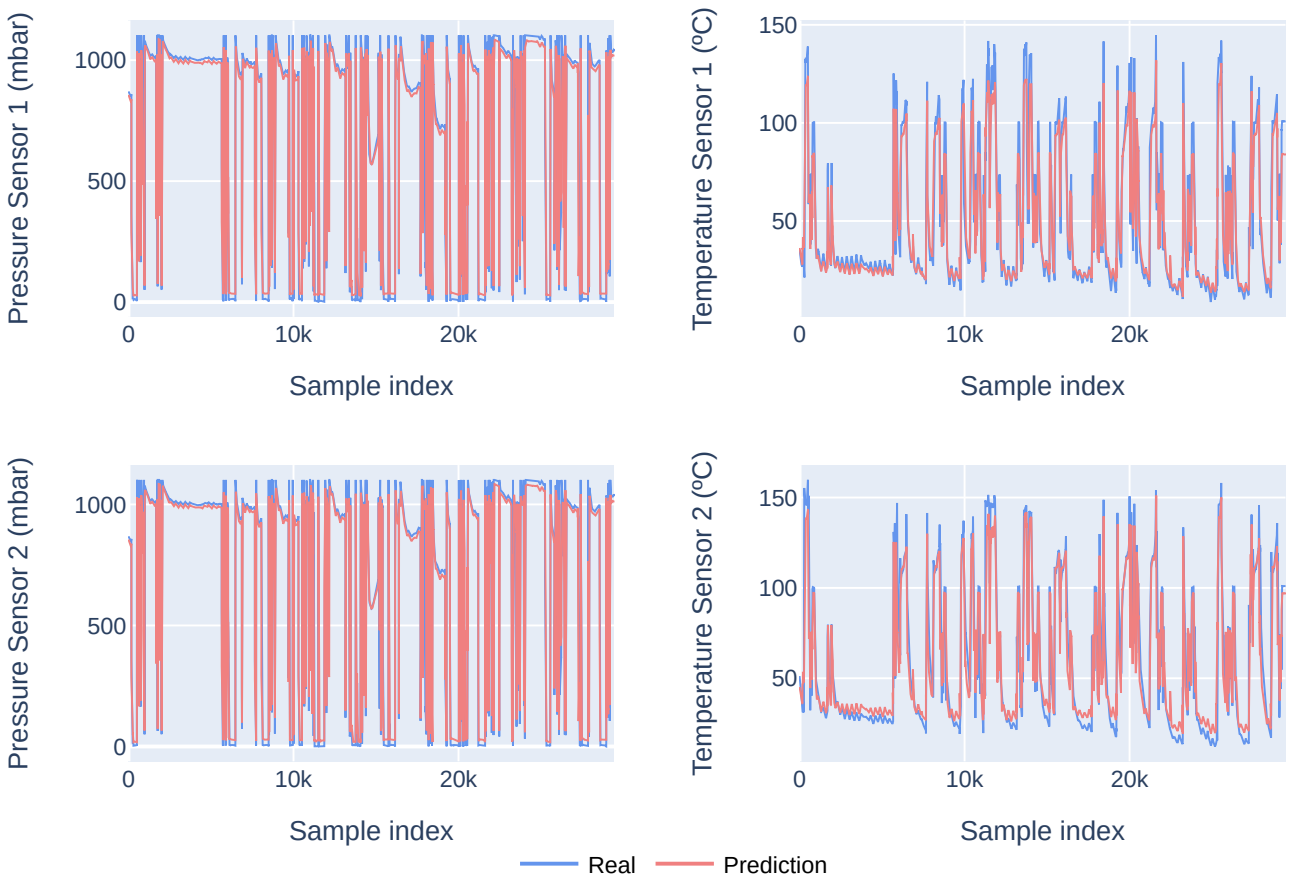

Fig. 4. Forecasting results for the different sensors.

- True Negative Rate (TNR), also known as specificity, measures the proportion of actual normal data points that are correctly classified as normal.
- The Geometric Mean (G-Mean) takes into account the relative balance of the classifier's performance on both classes and is defined as a function of the rate of TP and TN.
- Precision: measures the proportion of correctly classified anomalous data points out of all the data points predicted to be anomalous.
- Recall (also known as sensitivity): measures the proportion of correctly classified anomalous data points out of all the actual anomalous data points.
- F1-score: is the harmonic mean of precision and recall, and provides a balanced measure between them.
- Area Under the Receiver Operating Characteristic curve (AUC-ROC): measures the model's ability to distinguish between normal and anomalous data points, and provides a threshold-independent evaluation metric.
- Area Under the Precision-Recall curve (AUC-PR): measures the tradeoff between precision and recall and provides a threshold-independent evaluation metric.

Table II shows the obtained results for these metrics for a λ value of 0.5. Since our objective is to identify possible cyber-attacks through behavioural anomalies, it is crucial that the system detects all such events, even if it means allowing for some false positives. Therefore, the emphasis is on accurately detecting true positives, and the occurrence of false positives is acceptable as long as they are not too frequent. Considering this, our proposed methodology achieves a good balance between TPR and TNR, which translates into a high G-Mean value. A high AUC-ROC also implies a good threshold-independent performance. However, the main drawback of our current implementation is the high number of false positives, which implies a low precision and subsequent low F1-Score and AUC-PR results.

TABLE II
DETAILED EXPERIMENTAL RESULTS WITH THE PROPOSED LSTM MODEL

Metric	Result
TN	22997
FP	4983
TP	1419
FN	54
Precision	0.222
Recall	0.963
TPR	0.963
TNR	0.822
G-Mean	0.890
F1-Score	0.360
AUC-ROC	0.893
AUC-PR	0.593

V. Conclusions and Future Work

This paper showcases ongoing research that aims to enhance the security and resilience of industrial control systems. The proposed system is enabled through the use of LSTM neural networks to forecast future events and UEBA techniques to define entities of interest and generate behaviour models. The proposed solution has the potential to provide real-time detection of cyber and physical anomalies. We emphasize the advantages of UEBA in aggregating knowledge about the behaviour patterns of industrial devices to detect anomalous behaviours that may indicate potential security threats.

In Section IV, we discussed the proposed LSTM model performance while learning the trends and patterns in the data and predicting sensor values with a low error rate. Furthermore, we have carried out a preliminary validation of the anomaly detection capabilities of our model using metrics that are frequently used for binary classification, showing that the model is capable of detecting artificially inserted anomalies and achieving good G-Mean and AUC-ROC values.

However, the main remaining problem is the high false positive rate, which is very frequent in anomaly-based intrusion detection systems [37]. This problem still needs to be addressed so that incident inspection by human operators is feasible. Several techniques for alert correlation have been proposed [38]. Their main goal is to reduce a large portion of false positives by replacing the anomaly detector's output with an aggregate of its output on all similar events observed previously.

For an in-depth assessment of our methodology, we highlight the need for more detailed validation of the capabilities of the proposed methodology, as well as a comparison with other state-of-the-art solutions. We have identified established datasets commonly used in the literature, namely Secure Water Treatment [39] and Water Distribution (WADI) [40]. Both are based on a scaled-down version of a modern water treatment plant and consist of 51 features with clearly labelled attacks. They also provide information on the correspondence between the different processes, devices, measurement sensors and actuators involved, making it possible to apply UEBA behaviour modelling.

In addition to our proposed approach, other synthetic anomaly insertion techniques may be used. *Carmona et al.* [41] propose to inject simple single-point outliers in the time series. They use a simple method, similar to our proposal, that does not use the mean of the distribution as a starting point for anomalies. At a set of randomly selected time points, a spike is added (or subtracted) to the time series. The spike is proportional to the interquartile range of the points surrounding the spike location. On the other hand, [42] introduce a method for generating artificial anomalies in power and energy grids by leveraging real-world anomalies as a foundation. The technique involves defining distinct scenarios and specifying a duration parameter to generate synthetic anomalies that simulate a variety of actual occurrences. This allows validation of general characteristics of anomalies that should be detected: sudden changes in the location or scale of the series, interruption of seasonality, etc.

Future work also includes introducing improvements in the LSTM architecture. LSTMs have well-known limitations [43], like their inability to handle temporal dependencies that are longer than a certain amount of steps, where the network struggles to learn and generalize to new examples. In order to address them, further study in more robust architectures is needed. For instance, recent studies have shown that Bidirectional LSTM (Bi-LSTM) networks are able to detect and extract more time dependencies than unidirectional LSTMs and resolve them more precisely [44]. Bi-LSTM networks propagate the state vector not only in a forward pass but also in a reverse direction. The main advantage of this approach is that dependencies in both time directions are taken into account. Another improvement can be introduced by implementing attention mechanisms [45]. The main advantage of attention mechanisms in LSTM network architectures is that they allow the model to selectively focus on the most relevant parts of the input sequence when making predictions [46]. This can significantly improve the performance of the model, especially in tasks that require understanding long and complex sequences.

Acknowledgements

The research leading to these results has received funding from the "INNOTWIN" project under the Grant agreement number MIP-20211031 from the *Centro para el Desarrollo Tecnológico Industrial* (CDTI). Marta Sestelo acknowledges financial support from Grant PID2020-118101GB-I00 funded by Ministerio de Ciencia e Innovación (MCIN/ AEI /10.13039/501100011033).

References

[1] T. Alladi, V. Chamola, and S. Zeadally, "Industrial control systems: Cyberattack trends and countermeasures," *Computer Communications*, vol. 155, pp. 1–8, 2020, doi: 10.1016/j.comcom.2020.03.007.

[2] R. Khan, P. Maynard, K. McLaughlin, D. Laverty, and S. Sezer, "Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid," in *Proceedings of the 4th International Symposium for ICS & SCADA Cyber Security Research 2016*, ser. ICS-CSR '16. Swindon, GBR: BCS Learning & Development Ltd., 2016, p. 1–11, doi: 10.14236/ewic/ICS2016.7.

[3] N. Kshetri and J. Voas, "Hacking power grids: A current problem," *Computer*, vol. 50, no. 12, pp. 91–95, 2017, doi: 10.1109/MC.2017.4451203.

[4] Y. Maleh, "IT/OT convergence and cyber security," *Computer Fraud & Security*, vol. 2021, no. 12, pp. 13–16, 2021, doi : 10.1016/S1361-3723(21)00129-9.

[5] V. do Carmo C. de Vargas, L. F. Dias Lopes, and A. Mendonça Souza, "Comparative study of the performance of the cusum and ewma control charts," *Computers & Industrial Engineering*, vol. 46, no. 4, pp. 707–724, 2004, computers and Industrial EngineeringSpecial Issue on Selected papers form the 29th.International Conference on Computers and Industrial Engineering.

[6] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, vol. 22, pp. 949 – 961, 2017.

[7] G. Pannell and H. Ashman, "Anomaly Detection over User Profiles for Intrusion Detection," *Australian Information Security Management Conference*, 01 2010.

[8] M. A. Salitin and A. H. Zolait, "The role of user entity behavior analytics to detect network attacks in real time," in *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, 2018, pp. 1–5, doi:10.1109/3ICT.2018.8855782.

[9] G. Sadowski, A. Litan, T. Bussa, and T. Phillips. (2018) "market guide for user and entity behavior analytics". [Online]. Available: https://www.gartner.com/en/documents/3134524

[10] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, "Security information and event management (siem): Analysis, trends, and usage in critical infrastructures," *Sensors*, vol. 21, no. 14, 2021, doi: 10.3390/s21144759.

[11] L. L. Dhirani, E. Armstrong, and T. Newe, "Industrial iot, cyber threats, and standards landscape: Evaluation and roadmap," *Sensors*, vol. 21, no. 11, 2021, doi: 10.3390/s21113901.

[12] J. Kim, J.-H. Yun, and H. C. Kim, "Anomaly Detection for Industrial Control Systems Using Sequence-to-Sequence Neural Networks," in *Computer Security*, S. Katsikas, F. Cuppens, N. Cuppens, C. Lambrinoudakis, C. Kalloniatis, J. Mylopoulos, A. Antón, S. Gritzalis, F. Pallas, J. Pohle, A. Sasse, W. Meng, S. Furnell, and J. Garcia-Alfaro, Eds. Cham: Springer International Publishing, 2020, pp. 3–18.

[13] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, 2017, pp. 140–145, doi: 10.1109/HASE.2017.36.

[14] J. Wang, Y. Lai, and J. Liu, "Stealthy attack detection method based on multi-feature long short-term memory prediction model," *Future Generation Computer Systems*, vol. 137, pp. 248–259, 2022, doi: 10.1016/j.future.2022.07.014.

[15] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using lstm networks," *Computers in Industry*, vol. 131, p. 103498, 2021, doi: 10.1016/j.compind.2021.103498.

[16] M. A. Umer, K. N. Junejo, M. T. Jilani, and A. P. Mathur, "Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations," *International Journal of Critical Infrastructure Protection*, vol. 38, p. 100516, 2022, doi: 10.1016/j.ijcip.2022.100516.

[17] I. Ortega-Fernandez, M. Sestelo, J. C. Burguillo, and C. Piñón-Blanco, "Network intrusion detection system for ddos attacks in ics using deep autoencoders," *Wireless Networks*, Jan 2023, doi: 10.1007/s11276-022-03214-3.

[18] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.

[19] J. Liu, X. Song, Y. Zhou, X. Peng, Y. Zhang, P. Liu, D. Wu, and C. Zhu, "Deep anomaly detection in packet payload," *Neurocomputing*, vol. 485, pp. 205–218, May 2022, doi: "10.1016/j.neucom.2021.01.146".

[20] D. Myers, S. Suriadi, K. Radke, and E. Foo, "Anomaly detection for industrial control systems using process mining," *Computers and Security*, vol. 78, pp. 103–125, September 2018, doi: 10.1016/j.cose.2018.06.002.

[21] M. Elnour, N. Meskin, K. Khan, and R. Jain, "A Dual-Isolation-Forests-Based Attack Detection Framework for Industrial Control Systems," *IEEE Access*, vol. 8, pp. 36 639–36 651, 2020, doi: 10.1109/ACCESS.2020.2975066.

[22] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*, I. V. Tetko, V. Kůrková, P. Karpov, and F. Theis, Eds. Cham: Springer International Publishing, 2019, pp. 703–716.

[23] A. G. Martín, I. Martín de Diego, A. Fernández-Isabel, M. Beltrán, and R. R. Fernández, "Combining user behavioural information at the feature level to enhance continuous authentication systems," *Knowledge-Based Systems*, vol. 244, p. 108544, 2022, doi: 10.1016/j.knosys.2022.108544.

[24] F. Nocera, S. Demilito, P. Ladisa, M. Mongiello, A. A. Shah, J. Ahmad, and E. Di Sciascio, "A user behavior analytics (uba)- based solution using lstm neural network to mitigate ddos attack in fog and cloud environment," in *2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, 2022, pp. 74–79, doi: 10.1109/SMARTTECH54121.2022.00029.

[25] M. Shashanka, M.-Y. Shen, and J. Wang, "User and entity behavior analytics for enterprise security," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 1867–1874, doi: 10.1109/BigData.2016.7840805.

[26] F. Rashid and A. Miri, "User and event behavior analytics on differentially private data for anomaly detection," in *2021 7th IEEE Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, 2021, pp. 81–86, doi: 10.1109/BigDataSecurityHPSCIDS52275.2021.00025.

[27] S. Babu *et al.*, "Detecting anomalies in users-an ueba approach," in *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2020, pp. 863–876.

[28] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020, doi: https://doi.org/10.1016/j.physd.2019.132306.

[29] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 06, no. 02, pp. 107–116, 1998, doi: 10.1142/S0218488598000094.

[30] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[31] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[33] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73 – 101, 1964, doi: 10.1214/aoms/1177703732.

[34] M. Shcherbakov, A. Brebels, N. Shcherbakova, A. Tyukov, T. Janovsky, and V. Kamaev, "A survey of forecast error measures," *World Applied Sciences Journal*, vol. 24, pp. 171–176, 01 2013, doi: 10.5829/idosi.wasj.2013.24.itmies.80032.

[35] T.-W. Yoo and I.-S. Oh, "Time series forecasting of agricultural products' sales volumes based on seasonal long short-term memory," *Applied Sciences*, vol. 10, no. 22, 2020, doi = 10.3390/app10228169.

[36] R. Casado-Vara, A. Martin del Rey, D. Pérez-Palau, L. de-la Fuente-Valentín, and J. M. Corchado, "Web traffic time series forecasting using lstm neural networks with distributed asynchronous training," *Mathematics*, vol. 9, no. 4, 2021, doi : 10.3390/math9040421.

[37] E.-S. Apostol, C.-O. Truică, F. Pop, and C. Esposito, "Change point enhanced anomaly detection for iot time series data," *Water*, vol. 13, no. 12, 2021, doi: 10.3390/w13121633.

[38] M. Grill, T. Pevný, and M. Rehak, "Reducing false positives of network anomaly detection by local adaptive multivariate smoothing," *Journal of Computer and System Sciences*, vol. 83, no. 1, pp. 43–57, 2017, doi: https://doi.org/10.1016/j.jcss.2016.03.007.

[39] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Critical Information Infrastructures Security*, G. Havarneanu, R. Setola, H. Nassopoulos, and S. Wolthusen, Eds. Cham: Springer International Publishing, 2017, pp. 88–99.

[40] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "WADI: A Water Distribution Testbed for Research in the Design of Secure Cyber Physical Systems," in *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*, ser. CySWATER '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 25–28, doi: 10.1145/3055366.3055375.

[41] C. U. Carmona, F.-X. Aubet, V. Flunkert, and J. Gasthaus, "Neural contextual anomaly detection for time series," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, L. D. Raedt, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2022, pp. 2843–2851, doi: 10.24963/ijcai.2022/394.

[42] M. Turowski, M. Weber, O. Neumann, B. Heidrich, K. Phipps, H. K. Çakmak, R. Mikut, and V. Hagenmeyer, "Modeling and generating synthetic anomalies for energy and power time series," in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, ser. e-Energy '22. Association for Computing Machinery, 2022, p. 471–484, doi: 10.1145/3538637.3539760.

[43] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, nov 1997, doi: 10.1162/neco.1997.9.8.1735.

[44] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 3285–3292, doi: 10.1109/BigData47090.2019.9005997.

[45] A. Hernández and J. M. Amigó, "Attention mechanisms and their applications to complex systems," *Entropy*, vol. 23, no. 3, 2021, doi: 10.3390/e23030283.

[46] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," *Procedia CIRP*, vol. 99, pp. 650–655, 2021, doi: 10.1016/j.procir.2021.03.088.