Autoencoder-Based Detection of Insulin Pump Faults in Type 1 Diabetes Treatment

Elena Idi, *Graduate Student Member, IEEE*, Francesco Prendin, Giovanni Sparacino, and Simone Del Favero

***Abstract*—Individuals with type 1 diabetes (T1D) require lifelong insulin replacement to compensate for deficient endogenous insulin secretion, which would otherwise result in abnormal blood glucose levels. In recent years, significant investments have been made to improve T1D management, leading to the widespread adoption of accurate technology such as continuous glucose monitoring (CGM) sensors and automated insulin delivery systems. However, malfunctions in these devices, particularly pump systems, can cause undesirable interruptions of insulin delivery posing significant safety risks if not promptly addressed. Due to the low frequency of these episodes, developing accurate algorithms to identify insulin pump faults remains a challenge. To address these issues, this paper proposes a novel approach for detecting insulin pump faults (IPFs) by combining the ability of a long short-term memory (LSTM) autoencoder to extract features, with the strength of random forest to distinguish between anomalous and normal patterns. This method was developed and evaluated using data from 100 subjects, simulated over 90 days with the UVa/Padova T1D Simulator, an FDA-approved nonlinear computer simulator of T1D physiology. In the test set, the proposed algorithm identified the 93% of the total faults, while raising 2 false alarms in 3 months on average. These findings suggest that deep learning algorithms can enhance the safety and reliability of insulin pump systems, contributing to more effective therapeutic technologies.**

***Index Terms*—Fault detection, insulin pump, automatic feature extraction, insulin pump occlusions.**

I. Introduction

Type 1 Diabetes (T1D) is a chronic metabolic disorder that affects millions of individuals worldwide [1]. Because of the autoimmune destruction of the pancreatic beta cells, this condition leads to a fundamental impairment in the body's ability to regulate blood glucose levels, requiring lifelong insulin replacement therapy. The daily management of T1D is complex and highly demanding for individuals, which have to frequently monitor their glucose levels, modify lifestyle behavior and compute tailored insulin injections. While excessive insulin administration can lead to hypoglycemia (i.e., glucose levels below 70 mg/dL), with the potential to trigger seizures, comas, and even death [2], on the other side, insufficient insulin delivery could cause prolonged hyperglycemia (i.e., blood glucose levels exceeding 180 mg/dL). This condition is usually related to the development of chronic complications, including neuropathies, nephropathies, and cardiovascular diseases [3]. Furthermore, significant instances of hyperglycemia can also present an immediate risk to individuals, particularly when resulting from extended periods without insulin, leading to diabetic ketoacidosis (DKA), a life-threatening condition that requires patients' hospitalization.

In the last decades, technological advancements have significantly improved and revolutionized the management of T1D, introducing key innovative devices such as minimally invasive glucose sensors, known as Continuous Glucose Monitoring (CGM) sensors, that enable frequent blood glucose concentration measurements (at intervals of 1–5 minutes) without the need for finger pricks. Moreover, portable pumps, designed for continuous subcutaneous insulin infusion (CSII pumps), can facilitate the continuous insulin administration and allow the subjects to change insulin dosage throughout the day, also in real time [4]. Finally, closed-loop systems that adjust insulin infusion based on CGM readings have been developed and put on the market. These systems, usually referred to as artificial pancreas (AP), have shown clear potential to enhance the quality of glucose control while simultaneously reducing the actions requested to the patient to manage the disease [5].

Unfortunately, both CGM sensors and insulin pumps are susceptible to malfunctions, which can impact the overall functioning of the system and pose critical safety risks for the individual [6].

Insulin pump faults, such as infusion set blockages, mechanical failures and dislodgments, can result in undesirable insulin delivery interruptions to different degrees, including complete halts. If left untreated, these faults often lead to hyperglycemia and ketonemia [7], making pump issues one of the major contributors to diabetic ketoacidosis [8].

The early identification of IPF is a challenging task because insulin action on BG levels is not immediate and requires some time (∼45 minutes). For this reason, the rises in glucose levels related to the insulin absence (e.g. because of a pump occlusion), may become apparent on the glucose profile only some hours after the start of the malfunction. Nighttime occurrences of pump faults are particularly hazardous as individuals are often asleep and less prompt to intervention.

Received 5 October 2024; accepted 29 November 2024. Date of publication 16 December 2024; date of current version 6 February 2025. *(Corresponding author: Simone Del Favero.)*

The authors are with the Department of Information Engineering, University of Padova, 35131 Padova, Italy (e-mail: simone.delfavero@unipd.it).

Digital Object Identifier 10.1109/JBHI.2024.3518233

To deal with possible failures, some of the insulin pumps available in the market are indeed equipped with a self-monitoring system able to generate alerts to warn the patients when occlusions are suspected. Nevertheless, as reported in Gibney et al. [9], there are still multiple undetected episodes, called *silent occlusions:* in particular the authors reported that respectively one-third of the patients that inserted the set with the applicator and one-half of the patients that inserted the set manually, experienced undetected insulin pump malfunctions.

1) IPFs Detection, State of the Art: Given the practical importance of the automatic identification of faults that affect insulin pumps, a number of studies in existing literature have addressed this issue. In particular, the problem was investigated by developing: heuristic methods [10], model-based approaches [11], [12], [13], [14], [15] and machine learning-based strategies [16], [17].

Howsmon et al. [10] developed a heuristic approach that defines three fault indicators by leveraging glucose and insulin data. Then, when these indicators simultaneously exceed previously tuned thresholds, an alert is raised to warn the subject that an IPF is on-going.

Instead, in the model-based approaches presented in literature, the patient is intended as a dynamic system, where the output to control (i.e., glucose concentration) is affected by two inputs (insulin injection and meal intake). The model of the patient, that describes his physiology, is employed to predict future blood glucose levels, assuming that the insulin pump is properly working. Significantly large disparities between model predictions and CGM measurements indicate potential malfunctions. Different models can be used, including linear black-box models (as in [12], [13], [14], [15] and physiological non-linear models (as in [11]). As limitation, the performance of the model-based approaches is strongly dependent on the quality of the model of which they leverage, and accurate models of T1D physiology are nontrivial to identify.

To overcome this limitation, machine learning-based strategies [16], [17] have been developed. Both the approaches are based on the manual definition of features to detect anomalies in the system and are developed using in-silico data generated with the UVa/Padova T1D Simulator. Specifically, the main features employed are: i) the CGM signal, ii) the ratio between the trend of glucose concentrations and the carbohydrates on board (as in [18]), and iii) the ratio between insulin on board and carbohydrates on board. Concerning the machine learning models investigated, in [16], the authors employed multidimensional unsupervised anomaly detection techniques (i.e., Local Outlier Factor, Isolation Forest, and Histogram-based Outlier Scores). Of note, the proposed methods require a large amount of patient-specific data. Conversely, [17] examines both the previously mentioned unsupervised classifier and supervised classifiers. In accordance with the recent published guidelines [19], the approaches presented in [17] were developed in both population and personalized settings. Notably, while personalized strategies confirmed better performances, in case of limited patient-specific data the results suggested the adoption of population supervised classifier (with logistic regression and random forest as the two best options).

2) Paper Contribution and Organization: In this work, we addressed the problem of the real-time IPF detection by developing a deep learning approach based on a recurrent autoencoder for the automatic feature extraction, and a random forest classifier aiming at distinguishing the anomalous state of the faults from the normal condition. While these methodologies are becoming popular tools for the identification of faults into the industrial and manufacturing field [20], their use within the T1D research community is at its early stage.

The paper is organized as follow: Section II describes the dataset used and reports the details of the simulation, while the methodology is described in Section IV. The results achieved, in accordance to the criteria presented in Section III, are presented and discussed in Section V and VI. Finally, some conclusions are provided in Section VII.

II. Dataset

The pipeline proposed in this work is assessed using one of the latest versions of the UVA/Padova T1D Simulator [21], an FDA-accepted simulator of T1D physiology, developed for in silico testing prior to clinical trials.

Two synthetic datasets were generated, each consisting of 100 virtual subjects monitored over a period of 3 months. Everyday, three main meals were simulated, occurring at random times uniformly sampled within specific intervals: [06:30, 08:00] for breakfast, [11:30, 13:00] for lunch, and [18:00, 20:30] for dinner. The carbohydrate intake for each meal was uniformly sampled within the intervals [9–97g], [31–124g], and [28–140g], respectively. Basal insulin administration was regulated by a model predictive controller (MPC) [22], and insulin boluses were administered by the patient at each meal, based on estimated carbohydrate amounts.

This simulator version incorporates multiple sources of complexity such as the errors in carbohydrates estimation (as in [23]) and the intra-day insulin sensitivity fluctuations (as in [24]) for a more realistic scenario.

Blood glucose measurements, as well as insulin and meal data, were accessible at intervals of Ts $=$ 5 minutes, a typical sampling frequency in Automated Insulin Delivery systems. The CGM sensor's measurement error was characterized according to [25]. Moreover, in one of the two datasets generated, one nocturnal fault per month is simulated, in accordance to the frequency reported in [26]. Each fault occurs on one randomly chosen day and is simulated by suspending insulin delivery for 6 hours, from midnight to 06:00. After six hours, it is assumed that patients become aware of the malfunction and resume insulin administration through manual intervention. These implementation choices align with established practices in the field and have been widely adopted in previous studies (e.g.; [10], [15], [16]). A summary of the populations' characteristics, detailing body weight, age, percentage of time below range (i.e., below 70 mg/dL), in range (i.e., within [70–180] mg/dL) and above range (i.e., above 180 mg/dL) is reported in Table I.

The data presented in this section, were divided as follow: the dataset without faults was used for the training of the autoencoder, while the dataset with faults was subjected to a

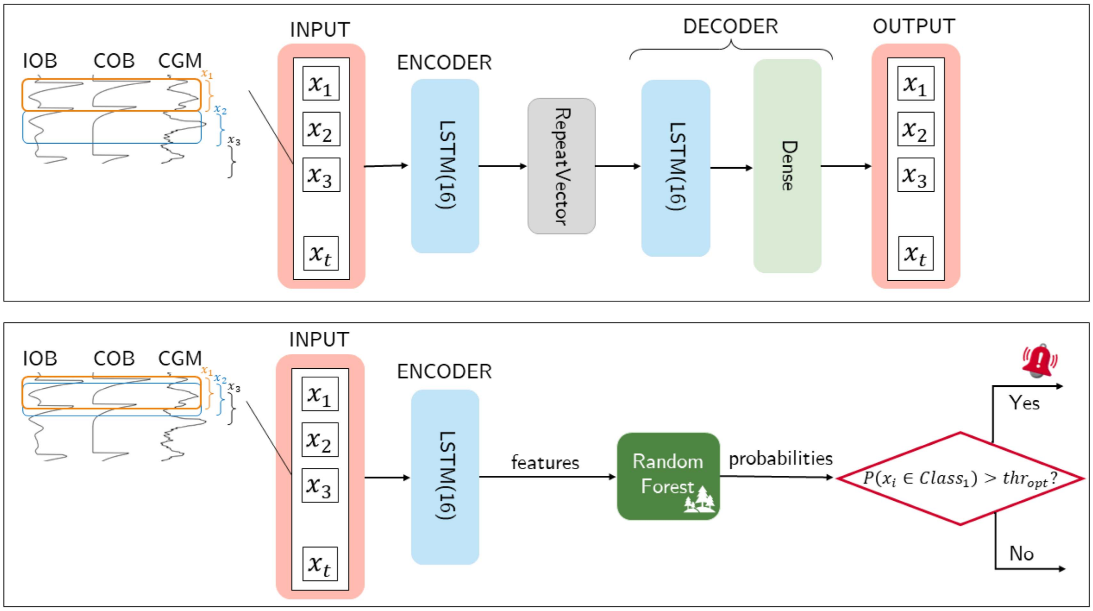

Fig. 1. In the upper panel is reported the autoencoder trained for the feature extraction. In the lower panel the pipeline for the detection is reported: in particular the encoder trained in used for the feature extraction and then Random Forest is applied for the classification and the alert generation.

TABLE I
DATASET SPECIFICS

Metric	Data without IPF	Data with IPF
Body Weight [kg]	75.2 (12.1)	75.2 (12.1)
Age [years]	33.8 (9.6)	33.8 (9.6)
Time below range (TBR) [%]	5.6 (5.2)	5.6 (5.2)
Time in range (TIR) [%]	76.1 (9.8)	75.1 (9.7)
Time above range (TAR) [%]	18.3 (9.6)	19.2 (9.4)

TABLE II
SUMMARY: DATASET PARTITIONING

Number of Subjects	Type of data	Step of the pipeline
Dataset1 100 subjects	No Faults	Autoencoder Training
Dataset2 80 subjects	1 IPF/month	Random Forest Training and Threshold Selection
Dataset2 20 subjects	1 IPF/month	Test of the Pipeline

training-test partitioning in 80:20 proportion for the application of the random forest (i.e., the algorithm was trained on 80 subjects and tested on the remaining 20). Since in real world scenarios training data might also be corrupted by anomalies, in a practical implementation, we envision to run preprocessing steps aiming to identify and discard all the data portion that can negatively affect the learning process of the autoencoder (AE). The summary of the data partitioning is reported in Table II.

III. EVALUATION CRITERIA

In accordance with [16], [17], the performance assessment relies on counting true positives (TPs), false negatives (FNs), and false positives (FPs). Specifically, if at least one alarm is triggered during an insulin pump fault, a TP is counted. Otherwise, a FN is recorded. Instead, a FP is considered if an alarm is erroneously raised in the absence of faults. Moreover, since a prolonged abnormal physiological state could persist for some hours after the restoration of insulin delivery, the alerts raised within a 6-hour window following insulin resumption (i.e., 6–12 hours from the start of the occlusion) are not counted as FPs.

Then, for each patient, we measured:

1) the recall (r), also known as sensitivity:

$$r = \frac{TP}{TP + FN}.$$

representing the fraction of IPFs correctly detected.

2) the number of false positive per day (fp/day).

3) the detection time, computed as the time elapsed from the start of the IPF to the generation of the alarms.

Then, the average metrics over the dataset are considered and are denoted as R, FP/day and $Delay$ in the following sections. It should be noted that, in this highly imbalanced dataset, where one fault arises per month (i.e. three faults for each subject in the whole simulation), true negative occurrences and their related metrics (e.g., specificity) are of limited interest [27].

IV. METHODS

A. Fault Detection Framework

The fault detection framework developed for IPFs detection combines the use of an autoencoder to automatically extract features from the data and a random forest for discriminating between anomalies and normal states. In particular, by learning the process to copy the inputs into the outputs, the autoencoder learns to compress and then reconstruct the input data, such that the hidden layers of the network, called *latent space*, can effectively capture the most relevant features needed for accurate reconstruction. In this way, the learned features can be seen as a representation of the underlying structure of the data and are employed as input to the random forest.

To this aim, we developed an ad-hoc detection pipeline, depicted in the lower panel of Fig. 1, which is composed by 3 main steps:

1) *Data Preparation:* this step consists of the pre-processing, windowing and normalization of the inputs, that are then used as inputs of the autoencoder.
2) *Automatic Feature Extraction using the Autoencoder:* once fed by the input sequence, the latent space representation of the input is expected to provide meaningful information in presence of an IPFs, which represents an abnormal situation.
3) *Anomaly Detection and Alert Generation:* the features extracted from the previous step are used as input of a random forest classifier to perform the classification in anomalous or normal samples. When an anomaly is detected, an alert is raised to warn the patient that an insulin pump fault occurred.

B. Step 1: Data Preparation

The available data, presented in Section II, includes readings from the CGM sensor and insulin injection amounts, obtained from the pump, as well as patient self-reported amount of CHO per meal. Since insulin boluses and meal intakes are impulse-like signals that can influence glucose levels for several hours, we introduce two new variables—Insulin on Board (IOB) and Carbohydrates on Board (COB). These variables, that have already been used in literature for several purposes (e.g. prediction [28], [29] and control [30]), carry information about the prolonged dynamics of insulin absorption and the gradual impact of carbohydrates on glucose levels, respectively.

In particular, the IOB represents an estimate of the amount of insulin injected and not yet absorbed and is computed as the convolution of the injected insulin with a suitable exponentially decaying function provided in Schiavon et al. [18]. Similarly, the COB represents an estimate of the carbohydrates that the patient has eaten but are not absorbed yet and it is computed as in [18]. Moreover, to mirror the physiological absorption time required by insulin and carbohydrates to impact on glucose levels, a delay is introduced in accordance with the literature [31] (∼45 and ∼15 minutes respectively).

Then, in the context of multivariate time series reconstruction, we derived input and output sequences using a sliding window approach with a fixed window size $L = 1$ hour (12 samples). Finally, the data are normalized using a min-max scaler at population level using the training data of the autoencoder (according to Table II, Dataset 1).

C. Step 2: Automatic Feature Extraction Using the Autoencoder

In fields such as health informatics, autoencoders (AEs) have proven effective for automatically generating feature set without human intervention. For example, AEs can generate features that are difficult to extract in medical imaging [32] or can highlight complex patterns in the EEG signals [33]. The overall architecture is usually composed of two parts: an encoder and a decoder. The encoder compresses the input data into a lower-dimensional representation (the *latent space*), while the decoder reconstructs the original input from this compressed representation. Specifically, at a generic time t, the input of the encoder part is the sequence of CGM, IOB and COB defined as

$$X_t^{enc} = [x_{t-L+1}, x_{t-L+2}, \ldots, x_t] \in R^{3\times L}$$

where $x_t = [CGM(t), IOB(t), COB(t)]$ and the output of the decoder is defined as

$$Y_t^{dec} = [\tilde{x}_{t-L+1}, \tilde{x}_{t-L+2}, \ldots, \tilde{x}_t] \in R^{3\times L}$$

where $\tilde{x}_t$ is the reconstruction of x_t.

When dealing with time series data, Long Short-Term Memory (LSTM) neural networks can be of help in efficiently learning and maintaining dependencies (both on long and short-term) from sequences of input data. These networks belong to the category of recurrent neural network (RNN), but they overcome the vanishing-exploding gradient related issues affecting deep RNN during the training. The forget, input, control and output gates are the key elements of the so-called memory cell of an LSTM and, at each time step, they control whether the incoming information is useful or if it must be erased from the cell. Therefore, an LSTM-based autoencoder represents a suitable approach for learning the complex dynamics characterizing the glucose-insulin system.

The architecture (summarized in the upper panel of Fig. 1) has been implemented in Python (Keras library [34]) and consists of different layers, each playing a specific role in the learning process. The encoder component employs a LSTM layer with 16 units, capable to learn the temporal dependencies within the input data. Moreover, to enhance model generalization and prevent overfitting, a dropout layer with a dropout rate of 0.3 is added after the encoder. Subsequently, a repeat vector layer is employed to replicate the encoded features across time steps, facilitating the decoder's ability to generate sequential outputs. Finally, the decoder, i.e., a LSTM layer with 16 units, is used for input reconstruction, together with a dense layer to restore the dimensionality of the reconstructed output to match that of the original input. The Mean Absolute Error (MAE) is adopted as loss function to minimize the differences between the reconstructed outputs and the original data for the training.

Once the AE has been trained, the decoder part is removed, leaving only the encoder LSTM-based model. This encoder is then used to transform the input sequences (i.e., CGM, IOB and COB) into a set of features, which represent the input data in a reduced-dimensional space. Figs. 2 and 3 (left panel) show the results at different steps of the detection pipeline: the output of the trained AE and the set of features extracted respectively.

D. Anomaly Detection and Alert Generation

The encoder designed in the previous step can automatically extract 16 features from the input sequences, which are then used as inputs of a Random Forest (RF) classifier [35] to distinguish between normal or anomalous states. Specifically, the RF is trained on a subset of 80 subjects who experienced 3 faults over 3 months of monitoring, with the aim to recognize the faulty samples among the data.

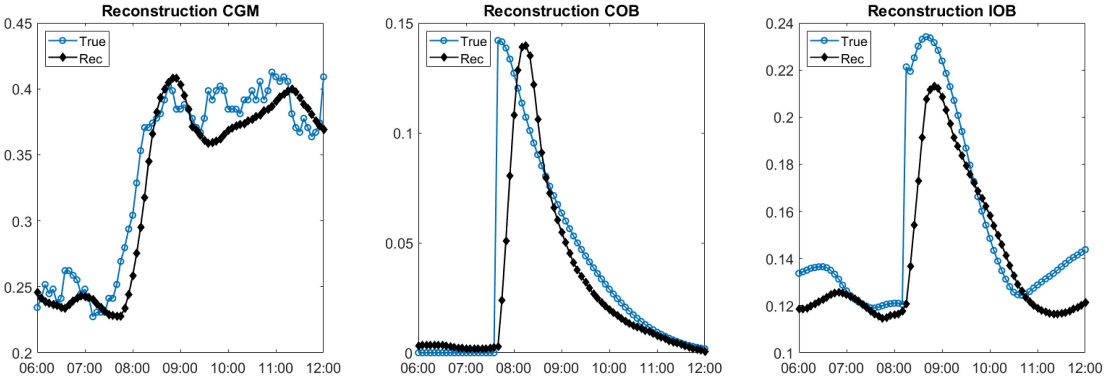

Fig. 2. Reconstruction of the inputs (CGM, COB and IOB respectively) on a training subject in absence of faults. The true signals are reported as blue solid line (circle markers), while the correspondent reconstructions are shown as linked black diamonds.

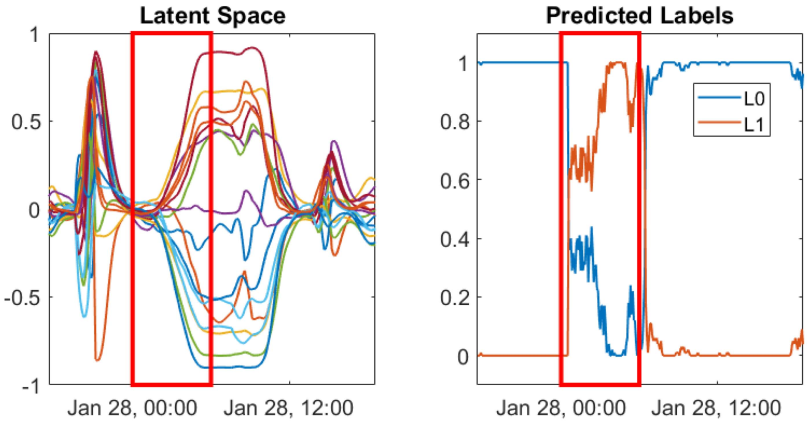

Fig. 3. In the left panel are reported the 16 features extracted with the encoder. The right panel shows the predicted probability computed by the Random Forest Classifier of belonging to that normal class ($L0$, in blue) or the anomalous one ($L1$, in orange). The red square indicates the occurrence of an insulin pump fault.

In this framework, the output of RF is a probability score for each class, which represents the likelihood that a given instance belongs to the faulty or non-faulty class. By default, such a classification decision is based on the class with the highest probability score. However, selecting a suitable threshold is crucial for achieving accurate performance. To this end, the classification threshold (thr_{opt}) was fine-tuned following the procedure described in [16]. Briefly, a grid of possible thresholds is defined and a cost function J is computed as

$$J(thr) = \sqrt{[1 - Recall(thr)]^2 + [FP/day(thr)]^2}.$$

J quantifies the distance from the point where the ideal performance are achieved ($Recall = 1$; $FP/day = 0$).

For each threshold tested, the average recall and FP/day in the training set and the corresponding value of J is computed and finally, the threshold that minimizes the cost function J is selected as optimal

$$thr_{opt} = argmin_{thr}\,(J(thr)) \qquad (1)$$

In this work, RF from the scikit-learn library [36] is used to compute the predicted class probabilities.

An example of the described probabilities is reported in the right panel of Fig. 3 during an insulin fault.

V. Results

Figs. 2 and 3 are generated to show the results at different steps of the detection framework. Particularly, Fig. 2 reported the output of the AE where the reconstructed inputs are shown together with the original signals during 6 monitoring hours of a subject in the training set. The AE seems effective in recreating the inputs: indeed, deviations from the original signal are expected (e.g., CGM data are corrupted by measurement noise) but limited, and the AE is able to replicate the core dynamics. This holds true also for the COB and IOB signals which seem to accurately mimic the physiological absorption curves of carbs and insulin. Moreover, while in Fig. 3 the left panel displays the set of features generated by the decoder, for a patient in the test set during an insulin pump fault, the right panel shows the probabilities calculated by the random forest of belonging to the normal class ($L0$) or the faulty one ($L1$). It can be seen that during a fault (highlighted with the red square), many of the extracted features spread towards the extreme values of 1 or -1. In accordance with that, the predicted probability of the data point to belong to the normal class decreases (from 1 to 0.4 and then to 0), while the probability of belonging to the anomalous class increases (from 0 to 0.6 and then to 1). A delay in the identification of the anomalous state is also visible in Fig. 3 (right panel): in particular, the predicted probability of belonging to the anomalous class ($L1$) increases

TABLE III
RESULTS EXPRESSED AS MEAN (STANDARD DEVIATION)

	Recall [ ]	FP/day [ ]	Delay [min]
AE	0.93 (0.17)	0.02 (0.04)	223 (67)

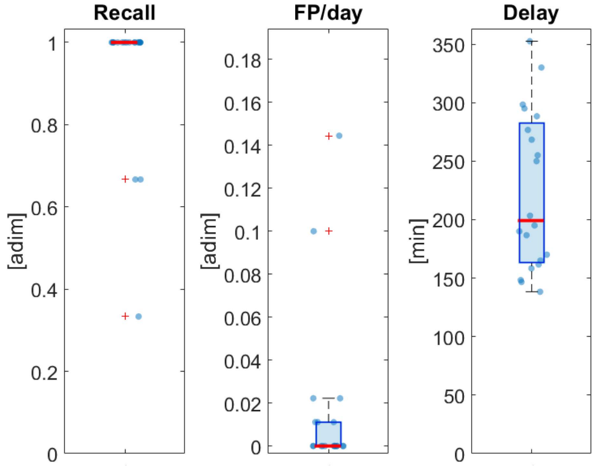

Fig. 4. Distribution of recall, FP/day and delay over the test set.

for some hours after the start of the pump occlusion. The delay depicted in the figure is reflected in the delay of the detection (reported in Table III), that represents the time required by the algorithm to recognize the fault. The detection results achieved on the test set after the computation of the threshold (on the training set) are summarized in Table III as mean and standard deviation. Moreover, the overall distributions of recall, FP/day and detection delay over the test set are reported in Fig. 4: a scatter plot is also superimposed to the boxplot, so that each blue dot represents the performance in one test subject.

In the test set, the algorithm shows promising performance by achieving a recall of 0.93, meaning that the architecture is able to recognize almost the totality (93%) of the insulin pump faults, while raising on average 0.02 false alarms per day, equivalent to less than 2 false alarms in 3 months. Furthermore, the delay of detection is computed as the time elapsed from the start of the insulin suspension to the raise of an IPF alarm. Thus, the algorithm requires $\sim$220 minutes on average to recognize the IPF.

VI. Discussion

The promising obtained results in the test set by the proposed approach underscore the ability of the latent space representation to effectively extract critical features that are indicative of potential issues with the insulin pump. In the left panel of Fig. 3, we presented the 16 features produced by the encoder during an episode of IPF. Most of these features exhibit a pronounced tendency to diverge rapidly towards extreme values when a fault episode occurs. This behavior suggests that the encoder is amplifying the anomalies in the data, particularly the deviations of input sequences from a normal situation. To better understand the role that these feature play on the following step (i.e.; RF classification), a feature importance analysis is conducted using the Mean Decrease in Impurity (MDI) method [37], which measures the importance of each feature based on how much it decreases the impurity in the RF model. In the context of RFs models, impurity measures how heterogeneous the data are at a given node. A high impurity means that the node contains a mix of data of different classes, while low impurity indicates that the data are more homogeneous, so that a node mostly contains data points of a single class. Since reducing impurity helps in making more accurate and reliable decisions in the model, Mean Decrease in Impurity method can provide insight on the importance of each feature in the classification process. Fig. 5 shows the extracted features ordered according to their value of MDI importance.

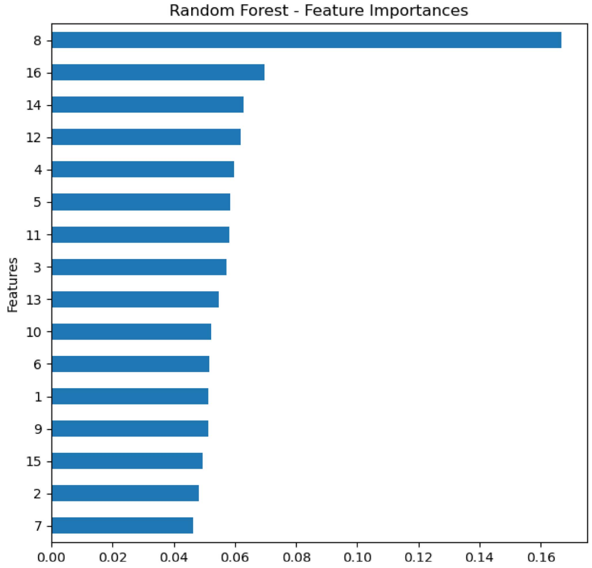

Fig. 5. Mean Decrease in Impurity (MDI) Importance. MDI importance value for each feature is shown in decreasing order. The larger the value, the more importance the model gives to that feature.

The top three identified features are also illustrated in Fig. 6. This figure presents a 3D scatter plot of the features $[f_8(t), f_{16}(t), f_{14}(t)]$ for a subject in the test set, showing distinct clusters for faulty periods (red crosses) and normal conditions (dark circles). It is worth noting that the AE is also able to distinguish not only between faulty and non-faulty periods, but it can provide further insights about the restoration phase (blue diamond) which represent the 6 hours after an IPF.

After the feature importance analysis, we investigate the performance achieved by the combination of the AE with the RF classifier in detecting the faults. The architecture achieves promising detection performances, as shown in Fig. 4. In fact, the algorithm scores a recall equal to 1 for all the subjects, with the exception of 3, and generates more than 0.1 false alarms per day for only one of the subjects in the test set.

Regarding the detection time, the recognition of the fault occurs in approximately 220 minutes on average. It's important to consider that, in the simulated scenario, the exact start of the fault is known and the impact of insulin absence becomes apparent on blood glucose only about 2 hours after the actual fault start. In view of this, the delay of detection obtained is in accordance with the study proposed in [38] by Klonoff et al. where they induced

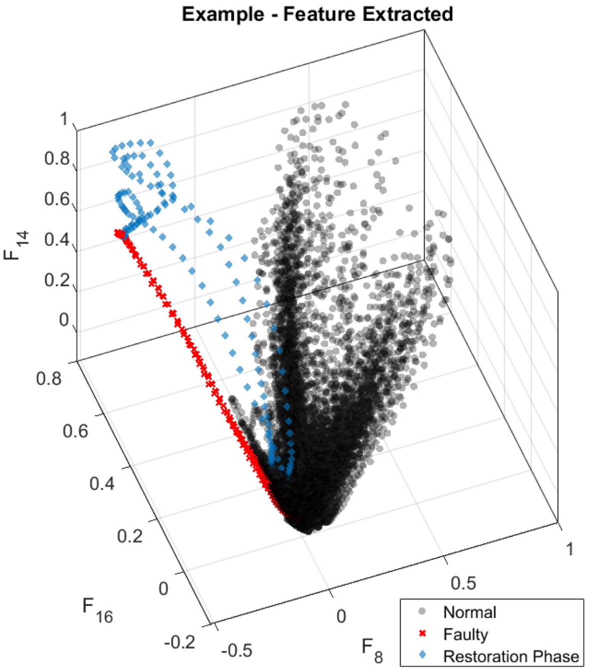

Fig. 6. 3D scatter plot of the features $[f_8(t), f_{16}(t), f_{14}(t)]$ of a subject belonging to the test set. The three features are reported as red crosses during an IPF, as dark circles in absence of faults and finally as blue diamonds during the restoration phase (the 6 hours after an IPF).

TABLE IV
RESULTS OF THE K-FOLD CROSS VALIDATION

Fold	Recall [ ]	FP/day [ ]
1	0.93 (0.17)	0.02 (0.04)
2	0.81 (0.20)	0.07 (0.10)
3	0.98 (0.07)	0.03 (0.06)
4	0.85 (0.26)	0.08 (0.10)
5	0.95 (0.22)	0.06 (0.09)
Average	**0.90**	**0.05**

real occlusions in clinical settings by applying a clip to the catheter of different insulin pumps available in the market and proved that the detection of the occlusion can take up to 4 hours.

To ensure a robust evaluation of the algorithm's performance across different subsets of the data, we employed a 5-fold cross-validation strategy. In this approach, the dataset is randomly partitioned into five equally sized folds and each fold is used as a test set once, while the remaining four folds are used for training. This process is repeated five times, with each fold acting as the test set exactly once. The results from each fold and the results averaged among the splits are presented in Table IV. Notably, the results across the different folds were consistent, indicating that the algorithm's performance is stable and not overly dependent on any specific subset of the data: our approach is able to recognize the 90% of the IPF on average while generating about 4 false alarms in 3 months.

Finally, the obtained results are compared with the ones reported in the literature, summarized in Table V.

The proposed approach outperforms the method proposed in [17] where RF is applied on a set of manually extracted features. In particular, the new architecture can recognize more faults with a lower generation of false alarms. Similar results in terms of recall are achieved by [15] but with a higher amount of false alarms raised, while similar amount of false alarms are generated with the approaches proposed in [16], [39] that instead exhibit a lower recall. It should be take into account that, while [39] used an older version of the simulator (generating a scenario potentially less realistic and challenging), this work and [15], [16], [17] leveraged the same version of the simulator and the settings were consistent across the environments, thus allowing for a fair comparison. Finally, the heuristic method proposed in [10] appears to achieve worse results in terms of both recall and FP/day: however, this method is developed and assessed on real-world data.

TABLE V
COMPARISON WITH THE STATE-OF-ART

Algorithm	Recall [ ]	FP/day [ ]	Dataset
AE-RF	**0.93**	**0.02**	Simulator v2018 [21]
Random Forest [17]	0.82	0.21	Simulator v2018 [21]
IForest [16], [17]	0.80	0.06	Simulator v2018 [21]
Manzoni et al [15]	0.91	0.12	Simulator v2018 [21]
Herrero et al [39]	0.80	0.08	Simulator v2014 [40]
Howsmon et al [10]	0.73-0.71	0.27-0.28	Real data

VII. CONCLUSION

Individuals with Type 1 Diabetes may potentially encounter a considerable safety risk if the insulin delivery is unexpectedly interrupted due to insulin pump faults or infusion set malfunctions. The automated identification of such issues can enhance the safety of T1D management systems, promoting trust toward therapeutic technologies.

In this work, we developed a novel approach that combines an LSTM-based autoencoder with a Random Forest classifier In particular, the AE was employed as an automatic feature extractor, that can capture the nonlinear relationships between the inputs (glucose concentrations, carbohydrates, and insulin administration). To do this, the AE was trained on faulty-free data to learn the essential features and then the encoder part was used for the automatic feature extraction in data containing faults. Finally, the features extracted from the latent space were used as inputs of a random forest classifier to distinguish patterns and anomalies in the data. The algorithm was developed and assessed using the UVa/Padova T1D Simulator, an FDA-accepted nonlinear computer simulator of T1D physiology. The obtained results are promising: almost the totality (0.93%) of the insulin pump faults is recognized, with less than 2 false alarms every 90 days. Despite employing one of the most realistic simulators available, simulated analysis are inevitably affected by simplifications. Therefore, the findings of this study should be validated through a larger dataset (possibly including other types of fault or different strategies for automatic control of insulin delivery), and in dedicated clinical trials, to test the robustness of the approach. Since the autoencoder is used as a black-box model, possible future studies could investigate the explainability of the latent space for a possible physiological interpretation of the features extracted. Furthermore, also the reconstruction errors of the inputs can introduce meaningful information for the detection of IPFs: even if the features extracted and the reconstruction errors are inevitably correlated, the quantification of the deviation of the reconstructed signals from the true ones can be of interest. Overall, the obtained results highlight the effectiveness of using deep learning techniques for IPF detection and underscore the potential of this approach to significantly improve the safety and reliability of T1D management systems.

References

[1] A. L. Notkins et al., “Autoimmune type 1 diabetes: Resolved and unresolved issues,” *J. Clin. Investigation*, vol. 108, no. 9, pp. 1247–1252, 2001.

[2] A. D. A. P. P. Committee and A. D. A. P. P. Committee:, “6. Glycemic targets: Standards of medical care in diabetes—2022,” *Diabetes Care*, vol. 45, no. Supplement_1, pp. S83–S96, 2022.

[3] D. Control, C. T. of Diabetes Interventions, and C. R. Group, “Retinopathy and nephropathy in patients with type 1 diabetes four years after a trial of intensive therapy,” *New England J. Med.*, vol. 342, no. 6, pp. 381–389, 2000.

[4] B. H. McAdams and A. A. Rizvi, “An overview of insulin pumps and glucose sensors for the generalist,” *J. Clin. Med.*, vol. 5, no. 1, pp. 5–22, 2016.

[5] H. Thabit and R. Hovorka, “Coming of age: The artificial pancreas for type 1 diabetes,” *Diabetologia*, vol. 59, no. 9, pp. 1795–1805, 2016.

[6] Y. Zhang, P. L. Jones, and R. Jetley, “A hazard analysis for a generic insulin infusion pump,” *J. Diabetes Sci. Technol.*, vol. 4, no. 2, pp. 263–283, 2010.

[7] D. Deiss et al., “Insulin infusion set use: European perspectives and recommendations,” *Diabetes Technol. Therapeutics*, vol. 18, no. 9, pp. 517–524, 2016.

[8] A. D. A. Dogan, U. L. Jørgensen, and H. J. Gjessing, “Diabetic ketoacidosis among patients treated with continuous subcutaneous insulin infusion,” *J. Diabetes Sci. Technol.*, vol. 11, no. 3, pp. 631–632, 2017.

[9] M. Gibney, Z. Xue, M. Swinney, D. Bialonczyk, and L. Hirsch, “Reduced silent occlusions with a novel catheter infusion set (BD FlowSmart): Results from two open-label comparative studies,” *Diabetes Technol. Therapeutics*, vol. 18, no. 3, pp. 136–143, 2016.

[10] D. P. Howsmon et al., “Continuous glucose monitoring enables the detection of losses in infusion set actuation (LISAs),” *Sensors*, vol. 17, no. 1, 2017, Art. no. 161.

[11] O. Vega-Hernandez, F. Campos-Cornejo, D. Campos-Delgado, and D. Espinoza-Trejo, “Increasing security in an artificial pancreas: Diagnosis of actuator faults,” in *Proc. 2009 Pan Amer. Health Care Exchanges*, 2009, pp. 137–142.

[12] A. Facchinetti, S. Favero, G. Sparacino, and C. Cobelli, “An online failure detection method of the glucose sensor-insulin pump system: Improved overnight safety of type-1 diabetic subjects,” *IEEE Trans. Biomed. Eng.*, vol. 60, no. 2, pp. 406–416, Feb. 2013.

[13] M. Cescon et al., “Early detection of infusion set failure during insulin pump therapy in type 1 diabetes,” *J. Diabetes Sci. Technol.*, vol. 10, no. 6, pp. 1268–1276, 2016.

[14] L. Meneghetti, A. Facchinetti, and S. Del Favero, “Model-based detection and classification of insulin pump faults and missed meal announcements in artificial pancreas systems for type 1 diabetes therapy,” *IEEE Trans. Biomed. Eng.*, vol. 68, no. 1, pp. 170–180, Jan. 2021.

[15] E. Manzoni, M. Rampazzo, A. Facchinetti, G. Sparacino, and S. Del Favero, “Monitoring statistical properties of Kalman filter residuals in an artificial pancreas to detect overnight pump malfunctions causing insulin suspension,” *Control Eng. Pract.*, vol. 141, 2023, Art. no. 105673.

[16] L. Meneghetti, M. Terzi, S. Del Favero, G. A. Susto, and C. Cobelli, “Data-driven anomaly recognition for unsupervised model-free fault detection in artificial pancreas,” *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 1, pp. 33–47, Jan. 2020.

[17] E. Idi, A. Facchinetti, G. Sparacino, and S. Del Favero, “Supervised and unsupervised approaches for the real-time detection of undesired insulin suspension caused by malfunctions,” *J. Diabetes Sci. Technol.*, 2024.

[18] M. Schiavon, C. Dalla Man, Y. C. Kudva, A. Basu, and C. Cobelli, “Quantitative estimation of insulin sensitivity in type 1 diabetic subjects wearing a sensor-augmented insulin pump,” *Diabetes Care*, vol. 37, no. 5, pp. 1216–1223, 2014.

[19] P. G. Jacobs et al., “Artificial intelligence and machine learning for improving glycemic control in diabetes: Best practices, pitfalls and opportunities,” *IEEE Rev. Biomed. Eng.*, vol. 17, pp. 19–41, 2024.

[20] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, “Autoencoder-based network anomaly detection,” in *Proc. 2018 Wireless Telecommun. Symp.*, 2018, pp. 1–5.

[21] R. Visentin et al., “The UVA/Padova type 1 diabetes simulator goes from single meal to single day,” *J. Diabetes Sci. Technol.*, vol. 12, no. 2, pp. 273–281, 2018.

[22] J. Pavan, D. Salvagnin, A. Facchinetti, G. Sparacino, and S. Del Favero, “Incorporating sparse and quantized carbohydrates suggestions in model predictive control for artificial pancreas in type 1 diabetes,” *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 2, pp. 570–586, Mar. 2023.

[23] C. Roversi, M. Vettoretti, S. Del Favero, A. Facchinetti, G. Sparacino, and H.-R. Consortium, “Modeling carbohydrate counting error in type 1 diabetes management,” *Diabetes Technol. Therapeutics*, vol. 22, no. 10, pp. 749–759, 2020.

[24] R. Visentin, C. Dalla Man, Y. C. Kudva, A. Basu, and C. Cobelli, “Circadian variability of insulin sensitivity: Physiological input for in silico artificial pancreas,” *Diabetes Technol. Therapeutics*, vol. 17, no. 1, pp. 1–7, 2015.

[25] A. Facchinetti, S. Del Favero, G. Sparacino, J. R. Castle, W. K. Ward, and C. Cobelli, “Modeling the glucose sensor error,” *IEEE Trans. Biomed. Eng.*, vol. 61, no. 3, pp. 620–629, Mar. 2014.

[26] A. C. van Bon, D. Dragt, and J. H. DeVries, “Significant time until catheter occlusion alerts in currently marketed insulin pumps at two basal rates,” *Diabetes Technol. Therapeutics*, vol. 14, no. 5, pp. 447–448, 2012.

[27] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” *PLoS One*, vol. 10, no. 3, 2015, Art. no. e0118432.

[28] J. Pavan et al., “Personalized machine learning algorithm based on shallow network and error imputation module for an improved blood glucose prediction,” in *KDH, ECAI*, 2020, pp. 95–99.

[29] M. F. Rabby, Y. Tu, M. I. Hossen, I. Lee, A. S. Maida, and X. Hei, “Stacked LSTM based deep recurrent neural network with Kalman smoothing for blood glucose prediction,” *BMC Med. Informat. Decis. Mak.*, vol. 21, pp. 1–15, 2021.

[30] S. D. Patek et al., “Modular closed-loop control of diabetes,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 11, pp. 2986–2999, Nov. 2012.

[31] S. Faccioli, A. Facchinetti, G. Sparacino, G. Pillonetto, and S. Del Favero, “Linear model identification for personalized prediction and control in diabetes,” *IEEE Trans. Biomed. Eng.*, vol. 69, no. 2, pp. 558–568, Feb. 2022.

[32] D. Ravì et al., “Deep learning for health informatics,” *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 4–21, Jan. 2017.

[33] T. Wen and Z. Zhang, “Deep convolution neural network and autoencoders-based unsupervised feature learning of EEG signals,” *IEEE Access*, vol. 6, pp. 25399–25410, 2018.

[34] F. Chollet et al., “Keras,” 2015. [Online]. Available: https://keras.io

[35] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

[36] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[37] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, “Understanding variable importances in forests of randomized trees,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, vol. 26.

[38] D. C. Klonoff, G. Freckmann, and L. Heinemann, “Insulin pump occlusions: For patients who have been around the (infusion) block,” *J. Diabetes Sci. Technol.*, vol. 11, pp. 451–454, 2017.

[39] P. Herrero et al., “Robust fault detection system for insulin pump therapy using continuous glucose monitoring,” *J. Diabetes Sci. Technol.*, vol. 6, no. 5, pp. 1131–1141, 2012.

[40] C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, and C. Cobelli, “The UVA/PADOVA type 1 diabetes simulator: New features,” *J. Diabetes Sci. Technol.*, vol. 8, no. 1, pp. 26–34, 2014.