Abstract—Intelligent Fault Detection (IFD), the use of machine learning-based methods and algorithms for the fault detection in modern systems becomes nowadays important due to the large number of data being generated by devices embedded in such systems. A typical example of such systems is Internet of Things (IoT)-based Cyber-Physical Systems (CPS) where IoT devices are used for better monitoring and control of such systems but at the same time due to their nature are susceptible to component faults. IFD depends on the number of data generated in such systems and their representation using system characteristics (features). Instance-based dataset reduction schemes used in Machine Learning (ML) aim to reduce the volume of data required during training while maintaining or preserving testing accuracy. Such reductions lead to less storage and processing time required for the trained models, which enables the use of lightweight IFD approaches in embedded devices found in the core of IoT-based CPS systems. In this work, we propose a machine learning-based framework for instance-based dataset reduction applied for IFD models. Our proposed framework is experimentally evaluated over two datasets. Results show that reduction is possible for up to 15.51% with an average accuracy improvement of 17% on the set of evaluated classification algorithms.

Index Terms—Intelligent Fault Detection (IFD), Cyber-Physical Systems (CPS), Machine Learning (ML), Internet Of Things (IoT)

I. INTRODUCTION

Cyber-Physical Systems, are combination of computational, networking and physical processes. In fact, the extensive integration of CPS in critical infrastructures elevated their role in ensuring economic development [1]. Hence, their resilience has become of utmost importance in all aspects of modern life. The proliferation of IoT technologies and their integration within CPS enables better monitoring, control and management of these systems. Constant communication of IoT devices enables the generation of a large amount of data in constant manner and thus machine learning-based approaches seems to be applicable. At the same time due the nature of integrated IoT devices, these systems are increasingly susceptible to component faults [2].

IFD refers to the use of machine learning methods for models used to detect faults. Such models are subject to data collection and instances representation in terms of system characteristics (features). Traditional artificial feature selection procedures [3] involve the use of machine learning methods to collect the features that represent better the instances of that dataset, while most recent approaches are focusing on the data collection procedure itself [4]. To the best of our knowledge there is no work that takes into consideration instance-based dataset reduction, a technique in machine learning that reduce the volume of data needed during training and lead to less storage and processing time required for the trained models, for IFD models.

Fault Detection in CPS has emerged as a challenging task due to the heterogeneity and large scale of such systems and the complexity of defining the faulty behaviour as this is a dynamic problem. Some approaches [5] utilise sensor and alarm data, characterising the new era of IoT-based CPS. Such IoT solutions for fault detection combine both machine learning approaches and human expertise while more recent approaches focus on the use of Deep Learning for fault detection. These procedures are based on two steps, the big data collection, referring to the generation of a large number of data using cloud-based solutions, and the deep learning based diagnosis and detection, that learns features from data and recognise healthy state of system [6].

Dataset reduction is a technique used in machine learning that aims to reduce the volume of data in a dataset. One way of doing this is by using instance-based dataset reduction. In this approach, dataset is reduced by removing instances that are used as part of the training and thus entail in reducing training time and computational resources required for the trained models while preserving or improving testing accuracy. An example of this approach is applied with instance-based classification algorithms, which perform their learning process at instance level. Those processes try to approximate the unknown function of the trained model by assigning the class labels to the actual instances and not violating model accuracy by removing redundant instances [7]. Another way of dataset reduction derives from the removal of unnecessary instances on algorithms that built decision boundaries (hyper planes), where instances removed are such that are away from the constructed boundaries, such as in Support Vector Machine (SVM) algorithm [8].

In this work we propose a generic ML-based dataset set reduction framework inspired from fault list reduction methodologies used in digital systems. More specifically, our framework accompanies techniques for features evaluation and ranking, such as Information Gain, with the notion of fault dominant. Instance-based dataset reduction proposed aim
TABLE I: Classification Matrix Example

| Actual Class | Predicted Class |  
|--------------|----------------|
| Normal       | a              |
| Faulty       | b              |

| Normal       | a              |
| Faulty       | c              |

| Faulty       | d              |

to reduce the volume of data needed during training while maintaining or increasing the testing accuracy. The main motivation of our framework is to derive a machine learning model for fault detection that will require less processing time and computational resources and thus being able to be applied in embedded devices found in the core of IoT-based CPS. Our proposed framework is evaluated through an IoT-based CPS simulated environment dataset [9] and a power system testbed model [10].

II. PROBLEM DEFINITION

We consider a dataset \( TR \) consisted of instances from the set of classes \( \{N, F_1, F_2, F_3, \ldots, F_n\} \) used for IFD models. Classes used are normal/healthy class \( N \) and a set of \( n \) abnormal faulty classes, each denoted by \( F_i \). Moreover, let a set of \( m \) machine learning classification algorithms \( Alg = \{A_1, A_2, A_3, A_4, \ldots, A_m\} \) that are used for IFD models based on \( TR \). Each model of an algorithm \( A_i \) has an accuracy, denoted by \( ac_{A_i}^{TR} \). Based on Table I, accuracy is computed by counting the correct number of predictions performed in the test phase of each model, that is trained in advance with a subset of instances of the initial dataset. \( a, d \) are the correct predictions performed by classifier and \( b, c \) the incorrect ones. Accuracy is a percentage metric (%) and is computed by \( ac = \frac{a+d}{a+b+c+d} \), \( ac_{A_i}^{TR} \) denotes the average accuracy achieved by the set of classification algorithms \( Alg = \{A_1, A_2, A_3, A_4, \ldots, A_m\} \) and models constructed based on \( TR \).

Problem II.1 (Dataset Reduction). Given dataset \( TR \) consisted of instances from the set of classes \( \{N, F_1, F_2, F_3, \ldots, F_n\} \) a set of classification algorithms \( Alg = \{A_1, A_2, A_3, A_4, \ldots, A_m\} \) with \( ac_{A_i}^{TR} \), we aim to find a reduced dataset \( TR' \) consisted of instances from the set of classes \( \{N, F_1, F_2, F_3, \ldots, F_n\} \) and \( n' \leq n \) such that for each algorithm \( A_i \) the \( ac_{A_i}^{TR} \geq ac_{A_i}^{TR'} \). This will entail into \( ac_{A_i}^{TR'} \geq ac_{A_i}^{TR} \).

Thus we aim, to find a reduction that maintain or increase the testing accuracy as per algorithm and average, by removing instances from the training dataset.

III. PROPOSED FRAMEWORK

Our proposed framework is consisted of three steps as illustrated in Fig. 1. This framework deals with datasets considering healthy (normal) and faulty (abnormal) instances. The proposed reduction is performed by removing instances from the faulty classes and thus reducing the size of the dataset. This is done with respect of the Problem II.1, for achieving higher per algorithm and on average accuracy. We now describe each step by also analysing its complexity based on the number of the faulty classes in the dataset \( n \).

1) Step 1: Features Ranking For Faulty Classes: In this first step we perform a feature analysis for the faulty classes by getting pairs of \((N, Fi)\) instances and forming a dataset. Then, we rank the features of each formed dataset using Information Gain ranking scheme. This rank allows the extraction of the most significant features that lead to fault detection of each faulty class \( Fi \). Information Gain, measures how much information a feature gives in respect of the predicted class. Information Gain is computed by

\[
InfoGain(Class, Feature) = H(Class) - H(Class|Feature)
\]

where \( H(Class) \) is the probability of an instance from the dataset to belong in the \( Class \) and \( H(Class|Feature) \) is a conditional probability used to describe the probability of an instance from the dataset to belong in the \( Class \) given that feature. This value is normalised between zero (0) and one (1).

The outcome of this ranking scheme analysis is a list of the most significant features of each faulty class \( Fi \), denoted by \( MSF_{Fi} \). This step complexity is \( O(n) \), as we aim to examine all the faulty \( n \) classes and obtain the according \( MSF_{Fi} \).

2) Step 2: Extraction Of Possible Dominant Relation Cases In this step the possible dominant relation cases as pair of faulty classes are extracted using the \( MSF_{Fi} \) lists for all the faulty classes, from Step 1. Possible dominant relation cases are pair of faulty classes that have at least half of their most significant features common and thus their coexistence in the dataset \( TR \) might not contribute to the improvement of the accuracy for the proposed models. Thus, the necessary condition for extracting these cases is to have two faulty classes \((Fi, Fj)\) have at least half of their most significant features common. Specifically, let \( CF_{i,j} \) denote the set of features common among the most significant between two faulty classes \((Fi, Fj)\). \(|CF_{i,j}|\) denotes the number of common features among the most significant as derived from previous step. This value must be greater than \( \text{ceil}(|MSF_{Fi}|/2) \). Then if \(|CF_{i,j}| \geq \text{ceil}(|MSF_{Fi}|/2) \) a possible dominant case exist between faulty classes \((Fi, Fj)\). This step complexity is \( O(n^2) \), as we need to examine all pairs of faulty classes.

3) Step 3: Experimental Evaluation Of Cases Using ML Algorithms: Based on the possible dominant relation
pair of classes derived from Step 2, in this step we define the dominant operator that allow us to determine whether a reduction of faulty instances is possible. Let \( TW_j \) being the dataset consisted of all the instances in \( TR \) except from the instances of \( F_j \) class. Let \( F_{TRAIN} \) being the set of classes without \( F_j \). Given a classification algorithm \( A_i \in Alg \) and \( ac_{TR}^{A_i} \) the accuracy derived by having all the faulty instances together as part of the training and \( ac_{TW_j}^{A_i} \) the accuracy of the same algorithm \( A_i \) with instances of \( F_j \) used only as part of the testing set then these class instances can be deducted from \( TR \) if \( ac_{TW_j}^{A_i} \geq ac_{TR}^{A_i} \). Specifically, the dominant operator \( \gg \) over the set of all classification algorithms is defined as:

Definition III.1 (Dominant Operator). Dominant operator \( \gg \) using a set of machine learning classification algorithms \( Alg = \{ A_1, A_2, \ldots, A_m \} \) is such that given datasets \( TR, TW_j \) and accuracies \( ac_{TW_j}^{A_i}, ac_{TR}^{A_i} \forall A_i \in Alg \):

\[
F_{TRAIN} \gg Alg F_j \text{ iff } ac_{TW_j}^{A_i} \geq ac_{TR}^{A_i} \forall A_i \in Alg
\]

In such case, a model for detecting faults using the set of algorithms in \( Alg \) can be trained using only \( TW_j \) instances but being able to detect faults in the system, with higher per algorithm and on average accuracy. For each pair of classes in a case the procedure described above is applied for each class, separately.

This step complexity is \( O(n^2) \) as the number of cases in terms of pairs can be up to \( n \times (n - 1) \).

IV. IoT EXPERIMENTAL SETUP AND RESULTS

We first evaluate our proposed framework in an IoT simulated environment, as it is described in [9]. In that work, authors consider an IoT enabled Energy Aware Smart Home (EASH). The communication environment of this system is simulated in OPNET simulator and communication is performed using Zigbee protocol. The topology described is a star topology, where peripheral monitoring elements report energy consumption measurements to a central coordinator every minute. The dataset consisted of normal, faulty and attack scenarios but for our evaluation we consider only the faulty scenarios instances. Faulty classes are F1: Low Energy Failure, F2: Routing Failure and F3: Packet Dropped Failure. We keep the same classification evaluation algorithms set as the one used for the experiments in that work, and experimental tool of Waikato Environment for Knowledge Analysis (WEKA) [11]. Thus \( Alg = \{ \text{NaiveBayes (NB)}, J48, \text{Multilayered Perceptron (MLP)} \text{ and Multinomial Logistic Regression (MLR)} \} \). In order to ensure that \( Alg \) set is representative enough we choose four different algorithms, which belongs to three separate categories: (Tree, Function, Probabilistic)-based classification algorithms. The baseline average accuracy \( ac_{TR}^{A_i} \) where \( TR \) considered of instances from the set of classes \( \{ N, F1, F2, F3 \} \) equals to 95.8%. Dataset contains 120 instances from which 48 are normal instances and each faulty class has 24 instances. Evaluation was performing using the percentage-split approach, thus we use 75% of the data as training data and the rest 25% as testing data. Below we explain how the steps of the proposed framework are applied using this dataset.

- **Step 1: Features Ranking For Faulty Classes**: Using the Information Gain, ranking scheme the most significant features of each faulty class are given in Table II. In this table we report the nine top features per class as this is the least common number of significant features of all faulty classes.

- **Step 2: Extraction Of Possible Dominant Relation Cases**: In order to derive the pair of classes that have at least half of their most significant features common. We then run a python script that takes into consideration all the combinations of features and classes defined above. The pair of faults that have at least half of their most significant features common and lead to possible dominant relation cases in this phase are: F1 & F2 \((\frac{5}{9})\), F1 & F3 \((\frac{5}{9})\) and F2 & F3 \((\frac{5}{9})\). For each pair an experimental case is derived.

- **Step 3: Experimental Evaluation Of Cases Using ML Algorithms**: Using the same classification algorithms as mentioned above for the set of \( Alg \), the evaluation cases derived from the previous step and experimental evaluation is described below. Results captured are presented in Figures 2, 3, 4. Baseline accuracy derived from the model constructed over dataset \( TR \) is given by blue colour bars in these charts. Note, that by leaving the same class out of training set accuracy is same and thus some bars show similar behaviour. The other two colours are commented below:

  - **Case #1: F1 & F2 (Fig. 2)**: With orange and green colour we present algorithms accuracies obtained after removing F1 and F2 instances from \( TR \), respectively. Thus, orange bars show accuracies over \( TW_1 \) and green bars over \( TW_2 \) experimental datasets. \( TW_1 \) models accuracy is 100% for all experimental algorithms, while in the scenario of \( TW_2 \) the accuracy drops from 94.4% to 75% in NB, remains 100% in J48, drops from 94.4% to 77.07% in MLP and from 94.4% to 75% in MLG. We conclude that \( F_{TRAIN} = \{ F2, F3 \} \) dominant F1 thus \( \{ F2, F3 \} \gg F1 \) exists as the accuracy is higher in all algorithms models.

  - **Case #2: F1 & F3 (Fig. 3)**: With orange and green colour we present algorithms accuracies obtained after removing F1 and F3 instances from \( TR \), respectively. Thus, orange bars show accuracies over \( TW_1 \) and green bars over \( TW_3 \) experimental datasets. \( TW_1 \) models accuracy is 100% for all experimental algorithms. In \( TW_3 \) dataset scenario accuracy is improved only in NB algorithm from 94.4% to 95.63%. For the rest of the algorithms accuracy drops from 100% to 75% in J48, from
TABLE II: Most Significant Features Per Class (IoT Dataset)

| CLASS | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|---|---|---|---|---|---|---|---|
| F₁  | ESI TRAFFICSENT APPL | ESI THROUGHPUT MAC | ESI DATA RECEIVED MAC | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL |
| F₂  | ESI TRAFFICSENT APPL | ESI THROUGHPUT MAC | ESI DATA RECEIVED MAC | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL |
| F₃  | ESI TRAFFICSENT APPL | ESI THROUGHPUT MAC | ESI DATA RECEIVED MAC | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL | ESI TRAFFICSENT APPL |

94.4% to 83.13% in MLP and from 94.4% to 93.75% in MLG. We conclude that $F_{TRAIN} = \{F2, F3\}$ dominant $F1$ thus, $\{F2,F3\} \gg_{Alg} F1$ exists as accuracy is higher in $TW_1$ models, similar as in the previous case (Case #1).

- **Case #3: F2 & F3 (Fig. 4):** With orange and green colour we present algorithms accuracies obtained after removing $F2$ and $F3$ instances from $TR$, respectively. Thus, orange bars show accuracies over $TW_2$ and green bars over $TW_3$ experimental datasets. In $TW_2$ models, accuracy drops from 94.4% to 75% in NB, remains 100% in J48, drops from 94.4% to 77.07% in MLP and drops from 94.4% to 93.75% in MLG. In $TW_3$ models, accuracy is improved from 94.4% to 95.63% in NB, drops from 100% to 75% in J48, from 94.4% to 83.13% in MLP and from 94.4% to 93.75% in MLG. Based on this results no dominant relation can be conducted from the scenarios of $TW_2$ and $TW_3$ datasets as the accuracy is not higher in all algorithms in either scenario of this case.

Based on the experiments performed above $TR$ dataset can be reduced into $TW_1$ by removing $F1$ instances as shown in Case #1, #2. The $a_{TR}$ with $TR'$ consisted of instances from the set of classes $\{N,F2,F3\}$ equals to 100.00%. As a result, accuracy improved on average of 4.2% with a dataset reduction of 20.00%.

V. POWER SYSTEM ICS EXPERIMENTAL SETUP AND RESULTS

We further evaluate our proposed framework over a dataset derived from an actual industrial control system testbed. This dataset is from a power system testbed modelled in Mississippi State University and Oak Ridge National Laboratory described in [10]. This dataset contains 37 scenarios that are used to model eight different faulty scenarios, normal operation and attack scenarios over a power system. This power system is consisted of two power generators G1 and G2 and Intelligent Electronic Devices (IEDs) that can switch the breakers of the generation on or off. For our experiments, we consider the dataset derived from the faulty scenarios that simulate and model single line to ground (SLG) faults over two transmission lines (L1&L2). Those faulty scenarios are short circuit faults (this is a short in a power line and can occur in various locations along the line, the location is indicated by the percentage range). Thus, faults considered are: $F1$: Fault from 10-19 on L1, $F2$: Fault from 20-79 on L1, $F3$: Fault from 80-90 on L1, $F4$: Fault from 10-19 on L2, $F5$: Fault from 20-79 on L2, $F6$: Fault from 80-90 on L2. Let, $TR$ being the dataset consisted of instances from the set of classes $\{N,F1,F2,F3,F4,F5,F6\}$. $Alg = \{NaiveBayes (NB), J48, Multilayered Perceptron (MLP) and Multinomial Logistic Regression (MLR)\}$. $a_{TR}$ using the same experimental parameters as in the IoT-based setup in the first experimental setup, we obtained a 78.065% base accuracy. The dataset is consisted of 9006 instances, from which 1712
are normal instances, 927 are F1 faulty instances, 1222 are F2 faulty instances, 1250 are F3 faulty instances, 1397 are F4 faulty instances, 1211 are F4 faulty instance and 1287 are F6 faulty instances. The same evaluation approach as in IoT experimental setup with the use of percentage split. 75% of the data was used as training data and 25% as testing data. Below we explain how the steps of the proposed framework are applied using this dataset.

- **Step 1: Features Ranking For Faulty Classes**: Features ranking for the faulty classes of the dataset is performed by using the Information Gain ranking scheme as we already define in the framework explanation section. For each class we keep the rank the ten most significant features and those are shown in Table III.

- **Step 2: Extraction Of Possible Dominant Relation Cases**: In order to derive the pair of classes that have at least half of their most significant features common we run a python script that takes into consideration all the combinations of features for the classes defined above. The pair of faults have at least of their most significant features common and lead to the possible dominant relation cases are: F1 & F4, F3 & F5, F2 & F6. For each pair an experimental case is derived.

- **Step 3: Experimental Evaluation Of Cases Using ML Algorithms**: Using the same classification algorithms as in the IoT experiments, the evaluation cases derived from the previous step and are experimental evaluated are given in this step below. Results captured are presented in Figures 5,6,7. Baseline accuracy derived from the model constructed over dataset TR is given by blue bar in these charts. Note, that by leaving the same class out of training set accuracy is same and thus some bars show similar behaviour. The other two colours are commented below:

  - **Case 1: F1 & F4** (Fig. 5): With orange and green colour we present algorithms accuracies obtained after removing F1 and F4 instances from TR, respectively. Thus, orange bars show accuracies over TW1 and green bars over TW4 experimental datasets. We see that in both reductions and models constructed over TW1 and TW4 datasets, accuracy is improved in NB, from 37.76% to 63.2% and 80.67% in the two scenarios, respectively. For J48 accuracy is also improved in both scenarios, from 96.8% to 97.46% and 98.78%. Using TW4 dataset accuracy is also improved in MLP and MLG from 83.43% to 87.01% and from 94.27% to 95.38% respectively. On the other hand, using TW1 accuracy is not improved in these two algorithms as it drops from 83.43% to 61.73% and from 94.27% to 90.71%, respectively. Based on these results we conclude that FTRAIN = \{F1, F2, F3, F5, F6\} dominant
framework achieved to improve the accuracy of the models, by removing instances that are redundant from the actual datasets. Specifically, an improvement of 12.4% on average with a dataset reduction up to 15.51% in the large dataset we examined for the power system testbed was achieved. Moreover, for the simulation IoT-based dataset reduction performed was of 20.0% and an average accuracy improvement of 4.2%. Such reduction enable the faster training models and the less storage required leaving space for lightweight model solutions. In future work, we aim to examine additional approaches for dataset reduction, considering the removal of instances without focusing on their class. Moreover, we aim to expand our framework and experiments in order to deal with other sources of abnormalities affecting such systems, as for example attacks and examine. Moreover, we aim to examine our framework ability over intrusion detection datasets.

VII. ACKNOWLEDGEMENT

This work has been supported by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE) and from of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

REFERENCES

[1] L. Shi, Q. Dai, and Y. Ni, “Cyber–physical interactions in power systems: A review of models, methods, and applications,” Electric Power Systems Research, vol. 163, pp. 396–412, 2018.
[2] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, “Efficient iot-based sensor big data collection–processing and analysis in smart buildings,” Future Generation Computer Systems, vol. 82, pp. 349–357, 2018.
[3] Y. Lei, M. J. Zuo, Z. He, and Y. Zi, “A multidimensional hybrid intelligent method for gear fault diagnosis,” Expert Systems with Applications, vol. 37, no. 2, pp. 1419–1430, 2010.
[4] Z. Duan, T. Wu, S. Gao, T. Shao, R. Malekian, and Z. Li, “Development and trend of condition monitoring and fault diagnosis of multi-sensors information fusion for rolling bearings: a review,” The International Journal of Advanced Manufacturing Technology, vol. 96, no. 1-4, pp. 803–819, 2018.
[5] D. Kim, S. C. Han, Y. Lin, B. H. Kang, and S. Lee, “Rdr-based knowledge based system to the failure detection in industrial cyber physical systems,” Knowledge-Based Systems, vol. 150, pp. 1–13, 2018.
[6] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, “Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data,” Mechanical Systems and Signal Processing, vol. 72, pp. 303–315, 2016.
[7] C. Kasemtaweewong and W. Suwannik, “Training set reduction using geometric median,” in 2015 15th International Symposium on Communications and Information Technologies (ISCIT). IEEE, 2015, pp. 155–156.
[8] R. Koggalage and S. Halgamuge, “Reducing the number of training samples for fast support vector machine classification,” Neural Information Processing-Letters and Reviews, vol. 2, no. 3, pp. 57–65, 2004.
[9] G. Tertytchny, N. Nicolaou, and M. K. Michael, “Differentiating attacks and faults in energy aware smart home system using supervised machine learning,” in Proceedings of the International Conference on Multi-Layer Intelligent Systems, 2019, p. 122–127.
[10] “Power system attack datasets - mississippi state university and oak ridge national laboratory.” [Online]. Available: http://www.ece.uah.edu/ thm0009/icsdatasets
[11] “Downloading and installing weka,” Machine Learning at Waikato University. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/download.html

F4 thus \{F1, F2, F3, F5, F6\} \supseteq_{Alg} F4 as the accuracy is higher in all experimental algorithms using TW.

- **Case 2: F3 & F5 (Fig. 6):** With orange and green colour we present algorithms accuracies obtained after removing F3 and F5 instances from TR, respectively. Thus, orange bars show accuracies over TW_3 and green bars over TW_5 experimental dataset. We observe that in both reductions and models constructed over TW_3 and TW_5 accuracy is improved in NB algorithm from 37.76% to 65.34% and 66.13% respectively. J48 model on TW_3 accuracy is improved from 96.8% to 97.23% but same algorithm model on TW_5 accuracy drops from 96.8% to 96.37%. For MLP and MLG algorithms accuracy is not improved in both dataset models as we observe a drop from 83.43% to 74.11% and 53.47% in MLP and from 94.27% to 89.7% and 91.14% in MLG for TW_3 and TW_5, respectively. No dominant relation can be derived in this case, as the accuracy is not improved in either scenario for all classification algorithms.

- **Case 3: F4 & F6 (Fig. 7):** With orange and green colour we present algorithms accuracies obtained after removing F4 and F6 instances from TR, respectively. Thus, orange bars show accuracies over TW_4 and green bars over TW_6 experimental datasets. We notice that in both reductions and models constructed over TW_4 and TW_6 datasets accuracy is improved in NB, from 37.76% to 66.1% and 65.55% and from 96.8% to 98.94% and 96.99% in J48, respectively. For the models of MLP and MLG over TW_4 and TW_6 datasets accuracy drops in both scenarios. Moreover, MLP models accuracy drops from 83.43% to 59.6% and 75.26% in TW_4 and TW_6 datasets, respectively, while in MLG accuracy drops from 94.27% to 91.82% and 90.46% in same datasets. No dominant relation can be derived in this case, as the accuracy is not improved in either scenario for all classification algorithms.

Based on the experimental evaluation performed above dataset TR can be reduced to TW by removing F4 instances (Case #1), as this was the only scenario that shows an accuracy improvement in all experimental algorithms. The acc^{TR}_TW with TR consisted of instances from the set of classes \{N, F1, F2, F3, F5, F6\} equals to 90.46%. Thus, reduction leads to an accuracy improvement of 12.4% on average with a dataset reduction of 15.51%. Having a bigger dataset we also observed that non functional-based optimisation algorithms (J48 and NB), are further improved by reduction schemes.

VI. CONCLUSION

This paper presents preliminary results on the dataset reduction framework proposed for the application domain of IFD in IoT-based CPS system datasets. Results show that proposed