NORTH SEA FLOW MEASUREMENT WORKSHOP
measuring for the energy transition

Data-driven Modelling for Condition-based Monitoring and Flow Regime Prediction in Flow Systems

Behzad Nobakht, TÜV SÜD National Engineering Laboratory
Yanfeng Liang, TÜV SÜD National Engineering Laboratory
Gordon Lindsay, TÜV SÜD National Engineering Laboratory
Sandy Black, TÜV SÜD National Engineering Laboratory

1 INTRODUCTION

Digitalisation strategies are now commonplace throughout manufacturing and engineering sectors. A major driver for this has been the fact that end-users now have a wealth of diagnostic data available to them from digital transmitters associated with a wide variety of devices installed throughout facilities. The data can be accessed in real time, for example, through OPC (Open Platform Communications) servers or stored in a database for future analysis. The vast amounts of data now being collected requires intelligent software to deliver analytics solutions and allow end-users to make better use of the data which they own.

The technical detail behind such strategies will vary between applications and organisations due to priorities, business needs and indeed due to differing interpretations of the word 'Digital'. However, for metrology purposes and in particular flow measurement, there are typically three key areas of interest to end-users: data analytics, condition-based monitoring (CBM) and predicitive analytics. In data analytics, both historical and real-time data can be analysed to uncover complex patterns and trends in primary and secondary flow measurement instrumentation and related back to physical processes and events. Through data-driven modelling, a faciliy's data can be used to replace inefficient 'time-based' calibration and maintenance schedules with 'condition-based' monitoring (CBM) systems which can remotely determine facility process conditions [1], detect fraudulent activity in custody transfer scenarios and meter calibration drift without the need for unnecessary manual intervention which is costly and time consuming for operators. In addition, specialised flow visualisation devices such as X-Ray tomography based systems can output data to which models can be applied to provide end-users with detailed insights as to the flow conditions within their multisensor system. Predicitive analytics embraces and expands upon the machine learning algorithms developed for CBM, and when fully realised and developed through the use of high resolution data sets allows end-users to forecast meter calibration requrements, erosion and corrosion [2] impact on flow meter functionality and even flow pattern development.

However, any flow application wishing to emabrace such concepts will be different with respect to physical layout, sensor availalblity and data resolution. This means that in order to develop CBMs and predicitive models which are useful and reflective of reality it is vital that experienced flow measurement engineers consult on model development and commissioning. In doing so, one effectively programmes into the model the collective experience of a facility's operators and its individual components.

Data acquisition in multisensor systems has grown into a significant source for diverse research and industrial areas, where mainly non-invasive and non-destructive examinations are needed [3]. An automated multisensor pipeline equipped with several sensors can output measurements simultaneously for industrial applications [3]. These sensors record key process factors in the form of both structured and semi-structured time series data. The data-driven models fed by these complex temporal data can then unleash different levels of information and aid in production optimisation.

In addition, a high resolution of faulty records allows the development of a reliable predictive model that can detect deviations from normal conditions, and if possible, identify the root cause of deviations. However, there might be practical limitations in generating real-world faulty events in flow measurement laboratories. Training data-driven models with insufficient data will lead to poor predictive performance, known as the data sparsity problem [4]

To overcome these limitations, synthetic data are widely employed to produce data that have not been observed in reality. Synthetic data preserve inter-relationships in data and reflect the statistical properties of real data to generate sufficient training data for data-driven algorithms.

Although many machine learning approaches have been used in multisensor systems, this field still faces many technical challenges [5]. Some challenges originate from measurement operations in continuous and uncontrolled environments, whilst some are unique to different sensors [3, 5]. Here we list several categories of challenges faced in data-driven modelling of flow measurement multisensor systems:

Data quality: Poor quality data displayed in the shape of missing values, highly correlated time series, duplicate data, numerous non-informative parameters, and outliers can cause significant challenges in the deployment of machine learning models in big data applications [6]. Statistical methods including imputation, outlier detection, dimensionality reduction, signal processing and data transformations enhance data quality.

Lack of standardised benchmarks for model evaluation: This is crucial when building a foundation for unsupervised/supervised data-driven diagnostics.

Feature extraction: Multisensor systems regularly generate a large amount of heterogeneous data in structured or unstructured formats. These data can contain complex inter-sensor relationships, time-dependent patterns and/or spatial correlations. Given the complexities in such multivariate data structures, it is hard to distinguish deviations from these relationships. Different conditions may have similar characteristics, making it challenging to build unique connections between features and conditions.

Computational costs: Training complex AI/deep-learning models for multi-dimensional data is resource-intensive and requires scale-up server configuration. This problem escalates in online learning systems, where models are updated continuously as more data streams become available [5].

Infrastructure: The data-driven approach from raw input data to predictive outputs requires a robust hardware infrastructure throughout the entire system's pipeline to perform high-speed processing of large volumes of information. This is due to the fact that intensive convolution operations and fully connected layers (vector-matrix multiplications) require efficient memory communication between both graphics processing units (GPUs) and central processing units (CPUs). In addition, optimisation strategies should be in place to distribute the workload of a program among different hardware resources [7].

Consequently, building such systems requires diverse skill sets, domain-specific knowledge, and a powerful processing unit to produce reliable analytics. The multivariate

time series models not only need to learn the temporal dependency in each variable but also require encoding the inter-correlations among different pairs of time series [8].

This paper therefore discusses research recently undertaken by TÜV SÜD National Engineering Laboratory (NEL) to develop application and device specific data-driven models which can provide accurate predictions as to the state of a given system. Two case studies are presented herein. The first details multiphase flow regime prediction by using probabilistic modelling of X-ray tomography data. The second focusses on the development of a data driven CBM system for a Coriolis-based metering system designed to detect undesirable operating conditions using real and synthetic data.

2 MULTIVARIATE TIME SERIES MODELLING

In univariate time series modelling, the goal is to predict a variable's behaviour exclusively based on information contained in its historical values. As a result, a univariate time series $X=\{x_1, x_2, \ldots, x_t\}$ can be described as a sequence of measurements (x_is) collected over time, where t signifies the length of the time series X. Conversely, multivariate class of models explain variations in a variable by referencing changes in current and past values of other variables.

There are now extensive applications of time-series modelling in modern industrial sectors, thanks to the wide range of sensors now available in instrumentation and machinery. For a classification task, the most intuitive feature extraction approach is to calculate the means, variances and other statistical properties of signals [9] and then use an algorithm such as Random Forest (RF) to predict classes. Basic signal statistics such as mean and standard deviations of time-series were previously used as selected features for activity recognition [10]. However, this simple approach has limited application in real-world scenarios where statistical features cannot represent the total characteristic of high dimensional, non-numerical, or seasonal data.

For many real-world problems, the data may represent a considerable temporal correlation, i.e. correlation within each record of a time series. Autoregressive (AR) models capture temporal correlation by employing a linear function of the previous observations plus random noise [11]. Since AR models predict each variable by all explanatory variables, including itself during previous time windows, they are a genuine starting point for the extraction of temporal dependencies and are already used broadly in applied research [11, 12]. However, these models are unable to handle a large amount of data with a high degree of nonlinearity or nonstationarity, because, autoregressive models, including nonlinear additive ones, make strong assumptions about stationarity and noise models in the time series [13, 14].

In a case where the characteristics of time series data change over a long period of time, or the input signals show unstable “noisy” behaviour, it will be difficult to estimate proper model parameters. This limitation indicates that the characterisation process of features by AR models can lead to poor predictive performance [13]. More importantly, the computational efficiency of such models is a significant barrier when applied in real-world scenarios [15]. In recent years, autoregressive models are embodied in Deep Neural Network models (DNNs) to circumvent the said limitations [16, 17]. In Deep AutoRegressive Networks (DARNs), successive deep hidden layers entail autoregressive connections, which allow for a fast solution to the model parameters [17].

2.1 Automated Multivariate Feature Learning

Multivariate time series learning becomes more challenging as the number of temporal variables that describe the data grows. In addition, conventional machine learning and

feature engineering methods are not powerful enough to capture the complex temporal and spatial patterns observed in multisensor data [18].

Progress in computational processing power and advances in Deep Learning algorithms have enabled automated feature encoding from the multivariate time series data. Different types of Neural Networks (NN) models have been widely used to learn temporal dynamics in a fully data-driven manner [18, 19]. Neural Networks can simultaneously extract information from multiple variables through multiple layers and identify associations between variables [19]. This advantage has led to a variety of state-of-the-art neural networks models that support both temporal and spatial guidance.

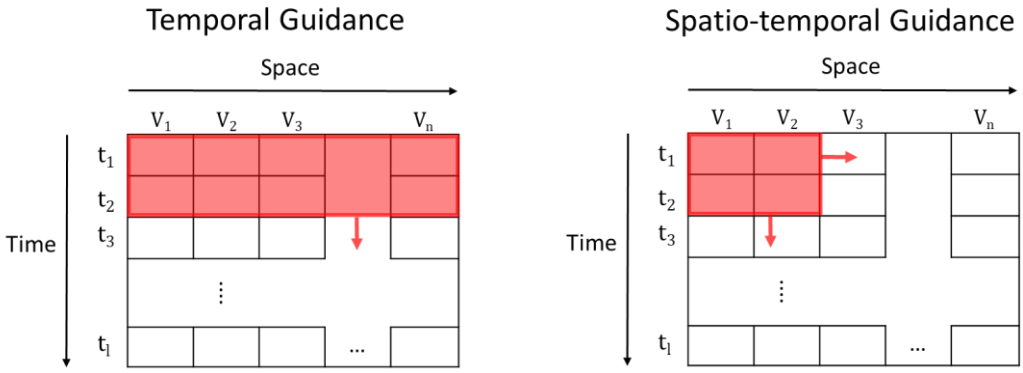

Fig. 1 - Temporal (left) and spatio-temporal (right) guidance in a 2D array of features V and time steps t

Recurrent Neural Networks (RNNs) can process a time series step-by-step to encode sequential information in the time domain [18]. Apart from time series forecasting, RNN were rarely employed in time series classification [20]. A type of RNNs, known as Long-Short Term Memory (LSTM) networks, can maintain information in memory for longer. Although LSTM can model temporally dependent data, the correlation between different subsamples may require a convolutional layer [20, 21].

Convolutional Neural Networks (CNNs) employ convolution operations by sliding filters across the time series where the filters work as a generic nonlinear transformation of a time series [22]. In 1D convolutional networks (Conv1D), the filters operate only on one dimension (time), ignoring spatial structure that might exist between features. In addition, Convolutional operations have a faster training time than LSTM [23]. 1D convolutions are also employed in Temporal Convolutional Networks (TCNs) to hierarchically capture relationships at different time scales [24]. The feature learning process in TCN first begins with low-level feature encoding using CNN that extract spatio-temporal information. Then, the encoded low-level features are passed to a classifier that learns high-level temporal characteristics. Thereore, TCNs present a unified approach to learn all two levels of information hierarchically.

A 2D convolution is required to learn both temporal and spatial structures. [25] introduced ConvLSTM layers for video-based action recognition. ConvLSTMs learn the internal representation of data through 2D convolutions while memorising short-term and long-term dependencies using LSTM cells. ConvLSTM networks hold convolutional structures which allow them to outperfom Fully-Connected LSTM neural networks [22, 25].

A 3D Convolutional layer can encode both temporal and spatial structures within the time series. [26] used 3D Convolutional neural networks (Conv3D) to capture motion information encoded in multiple adjacent frames. [27] proved that that 3D CNN can better extract spatio-temporal features than 2D CNN.

It is also possible to represent the inter-correlations among different pairs of time series in the form of signature matrices. Signature matrices evaluate the pairwise inner-product of two time series to capture shape similarities and correlations within each segment. Since turbulence at particular time series has little influence on the signature matrices, they are robust to signal noises. Previous studies by [8] and [28] demonstrated that calculating signature matrices for multivariate time series are crucial to characterise the system status at different scales.

In this work we extensively evaluated classification tasks by using only temporal guidance, spatio-temporal guidance without signature matrices, and spatio-temporal guidance with signature matrices.

3 CASE STUDY 1: FLOW REGIME PREDICTION

To identify flow structures, an X-ray 2D and 3-phase computed tomography (CT) device is used within NEL's multiphase facility for mixtures of gas, oil and water flows at low pressure. Then, fraction measurements of the 3-phase versus time are generated from the X-ray tomography device's digital output. The sum of the phase fractions must equal one, which therefore implies that in theory only two of the three phase fractions need to be measured. The project aimed to predict flow regimes based on time-varying fluid fractions of each X-ray tomography and void fractions throughout 32 sections of the pipe. The multivariate time series data are then fed to classification models.

In this study, we extensively studied the use of 3D convolutional neural networks (Conv3Ds) along with multi-scale signature matrices where convolutions infer both temporal and spatial characteristics from data. The learned features were then used to classify fluid flow regimes. The main objectives of this case study were:

1) Demonstrate the potential of Conv3D against current state-of-the-art approaches.

2) Highlight the importance of signature matrices in multivariate time series prediction.

3) Compare the impact of feature encoding by temporal guidance, spatio-temporal guidance, or no guidance.

3.1 Experiments

The data consisted of 182 samples of gas, oil and water fractions' time-series data for annular, bubble, intermediate, and stratified flow regimes. The instantaneous processed data over a 3-minute timeframe from the X-ray were collected for each sample. The test samples were taken at 40 Hz and made up of approximately 7000 time-steps (~3-minute timeframes).

To create an out-of-sample test set, we withheld the last 30 % (~ the last 54 seconds) of each sample aside and used it for the final evaluation of the models. This was a crucial consideration for testing the generalisability and performance of each model against an unseen test set. The remaining 70 % of data was used for training and validation such that training data to validation data ratio is 7 to 3.

Our data was imbalanced, meaning that it contained a different number of samples for each flow regime. For instance, there were more samples available for the intermediate flow regime than for the bubble flow regime. We made use of the Synthetic Minority Oversampling Technique (SMOTE) to generate synthetic samples for inadequate classes (i.e. oversampling) [29] to make sure that an equal number of flow regime samples were given to our models. After oversampling, we used a sliding window with 50 % overlap to

split up each sequence into subsequences of length 512. This gave us a total of 6508 training samples and 1532 test samples. Note that the test samples were kept aside before the oversampling was applied.

In this experiment, we examined four configurations in terms of temporal and spatial guidance:

(1) No guidance: We used an RF algorithm that ignored the temporal and spatial structures of multivariate time series, i.e., the expected model remained the same regardless of the order of features in temporal and spatial domains. The baseline RF model used four statistical properties including mean, standard deviation, skewness and kurtosis over the time domain for each time series.

(2) Only temporal guidance: Conv1D and TCN models were used to apply convolution only in the temporal domain. The models require 3D input of size $M*S*N$ where M is the number of samples in each batch, S is the length of sequences in each sample and N is the number of features.

(3) Spatio-temporal guidance using input signals: the models had to provide guidance in both the temporal and spatial dimensions. We used ConvLSTM and Conv3D models fed with the 5D input of size $M*R*L*N*C$ where R is the number of subsequences, L is the length of each subsequence, and C is the number of channels (C=1 in this case).

(4) Spatio-temporal guidance using signature matrices: The models used signature matrices to encode spatio-temporal correlation. We named these models TCN_sign ConvLSTM_sign and Conv3D_sign as they use TCN, ConvLSTM and Conv3D encoders, respectively. The models were fed with the 5D input of size $M*F*N*N*C$ where F is the number of frames, and C is the number of channels (C=3 in this case as signature matrices are calculated at three different scales).

In addition, all the neural network models used regularisations, including dropout, batch normalisation layers, maximum pooling and the hyper-parameters selected by optimal performance on the validation set. The RF model with no guidance could be used as a baseline model to evaluate the influence of encoding temporal and spatio-temporal information.

The Proposed model is a 3D Convolutional Neural Network fed with signature matrices (Conv3D_sign). The network inputs are concatenated signature matrices with three different lengths (w = 64, 32, 16) at each time step t. Initially, a 3D convolutional filter is applied to time frames (temporal domain) and signature matrices (spatial domain). Then, a series of regularisation techniques, including batch normalisation, maximum pooling layer, and spatial 3D version of dropout, are used to deal with overfitting. Then, the resulting output is given to another 3D convolutional layer, followed by batch normalisation, maximum pooling and spatial dropout. Finally, the last layer is a dense layer with softmax activation that predicts class distribution.

We trained the network using Adam optimisation with a batch size of 64. In all training phases, we used a validation set corresponding to 30 % of the training set. Therefore, the optimal values of the model's hyper-parameters were chosen based on the classification F1 score on the validation set. All the studied neural networks models have been implemented in the deep learning framework TensorFlow [30].

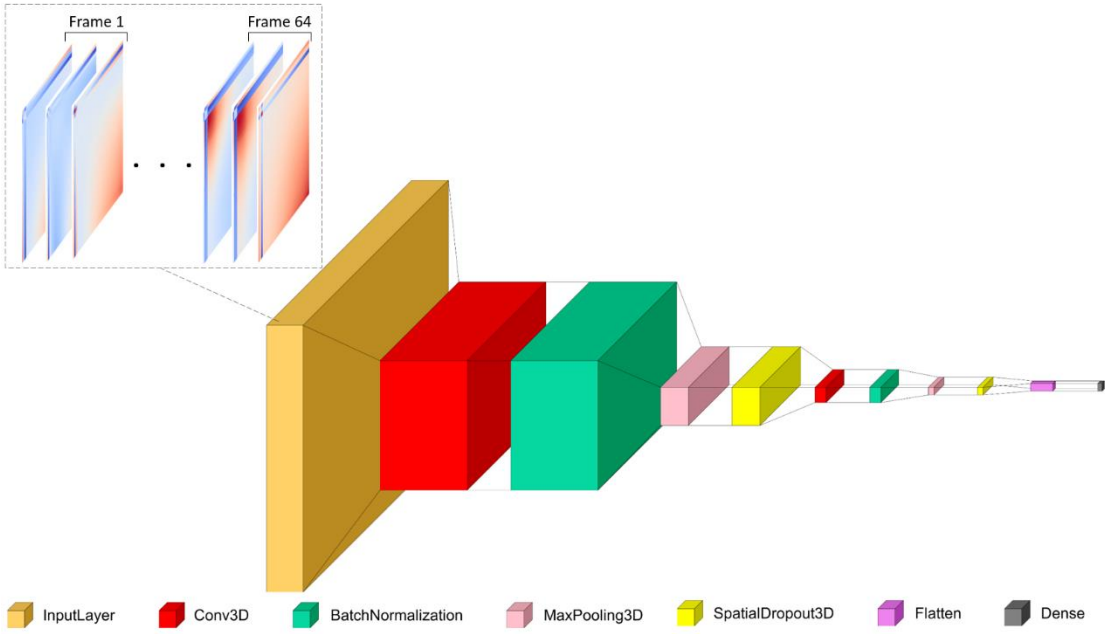

Fig. 2 – The proposed Conv3D_sign model: The network input is 64 frames of 3 signature matrices and the output is probabilities for each flow regime

Table 1 displays average model performance on the out-of-sample test set over five repeated experiments. Four evaluation metrics are reported, including F1 score, Cohen's kappa [31], recall and specificity, and corresponding standard deviations. Our baseline RF model had an average F1 score of 0.918, and a kappa value of 0.885. It is noteworthy that both Conv1D and TCN models with temporal guidance performed worse than the RF model, implying that by extracting basic signal statistics we better represent time-series features than by applying a 1D convolution to the temporal domain. Moreover, encoding spatio-temporal information through the Conv3D model improved the predictive power of the model. Finally, we observed a significant improvement in predictive power for models that use signature matrices and spatio-temporal guidance. On average, the models TCN_sign, ConvLSTM_sign and Conv3D_sign had a F1 score of 0.9558, 0.9258 and 0.9711, respectively. As can be seen from Table 1, our proposed Conv3D_sign model outperformed all other models by making the most of both spatial and temporal information within signature matrices.

TABLE 1 - Comparison of Predictive Performance, Average Score ± One Standard Deviation, for Case Study 1

Model	F1	Kappa	Sensitivity	Specificity	Guidance
RF	0.9184 ± 0.0028	0.885 ± 0.0038	0.9184 ± 0.0028	0.974 ± 0.0009	No Guidance
Conv1D	0.8629 ± 0.0149	0.8075 ± 0.0206	0.8629 ± 0.0149	0.9549 ± 0.0049	Temporal
TCN	0.8999 ± 0.0124	0.8569 ± 0.0175	0.8999 ± 0.0124	0.9645 ± 0.0047	Temporal
ConvLSTM	0.8902 ± 0.0162	0.8456 ± 0.0219	0.8902 ± 0.0162	0.9653 ± 0.0051	Spatial and Temporal
Coonv3D	0.9362 ± 0.0142	0.9093 ± 0.02	0.9362 ± 0.0142	0.9798 ± 0.0042	Spatial and Temporal
TCN_sign	0.9558 ± 0.0175	0.9367 ± 0.0251	0.9558 ± 0.0175	0.9855 ± 0.0065	Spatial and Temporal
ConvLSTM_sign	0.9258 ± 0.0404	0.8952 ± 0.0552	0.9258 ± 0.0404	0.9765 ± 0.0129	Spatial and Temporal
Conv3D_sign	**0.9711** ± 0.0109	**0.9583** ± 0.0159	**0.9711** ± 0.0109	**0.9904** ± 0.0049	Spatial and Temporal

In summary, our findings in this case study were as follows:

1. In multivariate time-series modelling, multi-scale signature matrices provided a substantial amount of information for both temporal and spatial feature learning.

2. 3D Convolutional neural networks outperformed current state-of-the-art approaches in multivariate time series classification.

3. Through comparing the impact of feature learning by temporal guidance, spatio-temporal guidance, or no guidance, we demonstrated the contribution of spatio-temporal characteristics in multi-sensor systems.

4 CASE STUDY 2: DATA DRIVEN MODELLING IN DETECTING AN ERROR IN A CORIOLIS FLOW METER

This case study aimed to demonstrate the necessity of synthetic data generation when faulty occurrences are expensive or impossible to obtain in real life. We attempted to anticipate real-world failures using synthetic faulty data to help data-driven models learn the system's normal behaviour and distinguish anomalous events.

The data consisted of 2,000 samples where each sample had 100 observations, resulting in a total of 200,000 observations with the data evenly divided into 2 classes – 'normal condition' and 'faulty condition'. The data from the normal class came from real field data output by a Coriolis flow meter during a normal 'healthy' operating condition. The data representing the 'faulty' condition consisted of synthetic data generated to mimic a specific recurring fault which was known to the operators and present within a limited sample set of real field data, which they were able to provide. Specifically, the fault condition resulted in a distortion in the inlet and outlet pick-up sensors of the Coriolis meter.

In order to find the optimal model in detecting a known error in a Coriolis flow meter, six classification models - random forest (RF), stochastic gradient boosting (GBM), decision tree (Tree), naïve bayes classifier (NB), neural network (NN) and k-nearest neighbours (KNN) were trained and built using real and synthetically generated faulty data.

The data was divided into 70 % training data, through which the models learnt the patterns, trends and correlations within variables associated with 'normal' and 'faulty' conditions. The models' prediction capability was then tested against the remaining 30 % of the data. Fig. 3 illustrates the performance of the models based on the training data measured against the range of Cohen's kappa values. It was observed that RF and GBM performed best by obtaining an average kappa value of 0.97 and 0.96 respectively. Note that the range of kappa value was obtained by retraining the models 5 times with random starting points. Each model's hyperparameters were selected by maximising optimal performance on the validation set. The black dot in Fig. 3 represents the median kappa value.

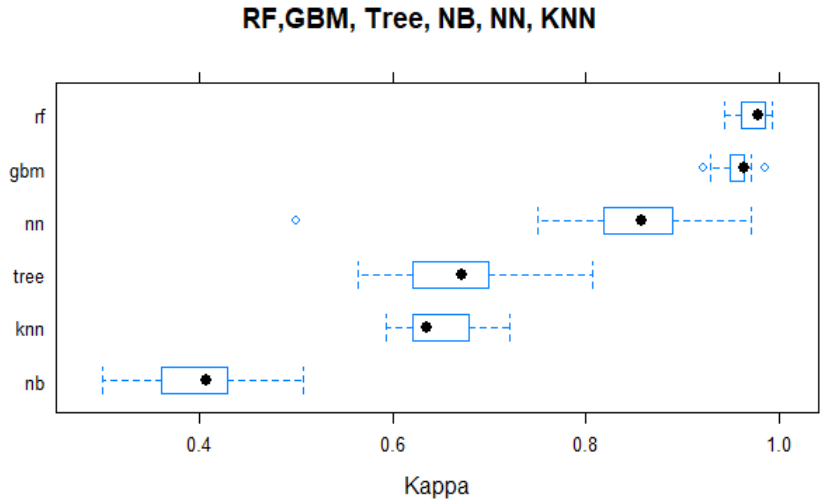

Fig 3: Kappa values for different models based on training data.

To validate the model's performance, each model was then tested using the 30 % validation data, where the performance of the model was measured by various metrics. From the results in Table 2, the optimal models based on the validation data were RF, with high scores in all metrics, followed closely by GBM. The worst model was NB with a sensitivity rate of 0.53, a kappa value of 0.447 and an F1 score of 0.657. However, it is interesting to note that NB had achieved a high specificity rate of 0.92. In other words, the model performed well in assigning all 'normal condition' data into the correct group, but failed to perform well in detecting the 'faulty condition'. The optimal model being defined as the one which performs well across the differing metrics.

TABLE 2 - Comparison of Predictive Performance Against Synthetic Magnet Validation Data: The Values Represent Average Score ± One Standard Deviation

Model	F1	Kappa	Sensitivity	Specificity
RF	**0.987** ± 0.001	**0.973** ± 0.002	**0.987** ± 0.001	**0.987** ± 0.002
GBM	0.978 ± 0.002	0.957 ± 0.005	0.973 ± 0.004	0.983 ± 0.002
Tree	0.819 ± 0.010	0.653 ± 0.010	0.783 ± 0.014	0.870 ± 0.030
NB	0.657 ± 0.010	0.447 ± 0.010	0.530 ± 0.010	0.920 ± 0.010
NN	0.933 ± 0.007	0.87 ± 0.017	0.910 ± 0.013	0.960 ± 0.027
KNN	0.831 ± 0.001	0.683 ± 0.002	0.777 ± 0.002	0.907 ± 0.001

The trained classification models were based on synthetically generated faulty conditions. To ensure that the models have the capability to be extended and applied in real-life situations to detect real faulty conditions, the models were then further tested on a set of data logged from a Coriolis meter where the meter was known to be exposed to the faulty condition for a period of time. The data from Coriolis meter contained 9,680 observations, 66 % of which were logged when the Coriolis meter was known to be exposed to an error. Each of the models was then used to detect the error with the prediction results given in Table 3.

TABLE 3 - Comparison of Predictive Performance Against Coriolis Meter's Data: The Values Represent Average Score ± One Standard Deviation

Model	F1	Kappa	Sensitivity	Specificity
RF	0.660 ± 0.024	0.383 ± 0.057	0.500 ± 0.023	**0.970** ± 0.025
GBM	**0.954** ± 0.025	**0.860** ± 0.059	0.969 ± 0.028	0.879 ± 0.013
Tree	0.800 ± 0.034	0.040 ± 0.014	0.996± 0.003	0.030 ± 0.010
NB	0.815 ± 0.030	0.154 ± 0.010	0.996± 0.003	0.121 ± 0.012
NN	0.821 ± 0.015	0.191 ± 0.083	**0.997** ± 0.002	0.152 ± 0.010
KNN	0.811 ± 0.020	0.246 ± 0.020	0.938 ± 0.010	0.273 ± 0.010

From the results shown in Table 3, GBM performed the best in successfully detecting the error within a Coriolis meter with a high F1 score of 0.954, and a kappa value of 0.86. Despite the fact that NN, Tree and NB had a sensitivity rate close to 1, they performed poorly in assigning normal condition data into the correct group as indicated by their low specificity rates. Consequently, GBM was chosen to be the optimal model, performing well across all metrics in detecting synthetically generated faulty data and real faulty data. The results are promising as despite the fact that the model was built using synthetically generated faulty data, it demonstrated the capability of extending its predictive power in detecting real faulty data which was output from a Coriolis flow meter during operation.

5 CONCLUSION

In this study, two significant problems in the real-time monitoring of the process industry were addressed: Multivariate time-series feature encoding and data sparsity.

Different scenarios were examined, to explore the impact of adding temporal and spatial guidance to the feature learning process. It was also demonstrated that multi-scale signature matrices could provide a substantial amount of information for feature learning. In addition, 3D Convolutional neural networks were able to outperform current state-of-the-art approaches in multivariate time series classification.

The enhancement in data-driven models' predictive power when using synthetic data was highighted alongside an illustration of how training a model with synthetically generated faulty events could improve data-driven models' predictive power in CBM systems.

6 NOTATION

X	Input time series
M	Number of samples in each batch
S	Length of sequences in each sample
N	Number of features
R	Number of subsequences
L	Length of each subsequence
C	number of channels
F	Number of frames

7 REFERENCES

[1] NOBAKHT, B. 2019. Probabilistic modelling of flow regimes using X-ray tomography data. NEL Report No. 2019_ 812.

[2] LIANG, Y. 2020. Predicting the remaining useful life of flow meters during erosive flow conditions. NEL Report No. 2020_435.

[3] DOBMANN, G., KURZ, J., TAFFE, A. & STREICHER, D. 2010. Development of automated non-destructive evaluation (NDE) systems for reinforced concrete structures and other applications. *Non-destructive evaluation of reinforced concrete structures.* Elsevier.

[4] HASSALL, K. L., WHITMORE, A. P. & MILNE, A. E. 2019. Accounting for data sparsity when forming spatially coherent zones. *Applied mathematical modelling,* 72**,** 537-552.

[5] CHEN, K., ZHANG, D., YAO, L., GUO, B., YU, Z. & LIU, Y. 2021. Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges, and Opportunities. *ACM Computing Surveys (CSUR),* 54**,** 1-40.

[6] GUDIVADA, V., APON, A. & DING, J. 2017. Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software,* 10**,** 1-20.

[7] DOMÍNGUEZ, J. M., CRESPO, A. J. & GÓMEZ-GESTEIRA, M. 2011. Optimization strategies for parallel CPU and GPU implementations of a meshfree particle method. *arXiv preprint arXiv:1110.3711*.

[8] ZHANG, C., SONG, D., CHEN, Y., FENG, X., LUMEZANU, C., CHENG, W., NI, J., ZONG, B., CHEN, H. & CHAWLA, N. V. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 1409-1416.

[9] DERLER, P., LEE, E. A., TRIPAKIS, S. & TÖRNGREN, M. Cyber-physical system design contracts. Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems, 2013. 109-118.

[10] VEPAKOMMA, P., DE, D., DAS, S. K. & BHANSALI, S. A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities. 2015 IEEE 12th International conference on wearable and implantable body sensor networks (BSN), 2015. IEEE, 1-6.

[11] DABLANDER, F., RYAN, O. & HASLBECK, J. M. 2020. Choosing between AR (1) and VAR (1) models in typical psychological applications. *PloS one,* 15**,** e0240730.

[12] BRINGMANN, L. F., VISSERS, N., WICHERS, M., GESCHWIND, N., KUPPENS, P., PEETERS, F., BORSBOOM, D. & TUERLINCKX, F. 2013. A network approach to psychopathology: new insights into clinical longitudinal data. *PloS one,* 8**,** e60188.

[13] BAI, Y., JIN, X., WANG, X., SU, T., KONG, J. & LU, Y. 2019. Compound autoregressive network for prediction of multivariate time series. *Complexity,* 2019.

[14] CHEN, R., LIU, J. S. & TSAY, R. S. 1995. Additivity tests for nonlinear autoregression. *Biometrika,* 82**,** 369-383.

[15] MINNEN, D., BALLÉ, J. & TODERICI, G. 2018. Joint autoregressive and hierarchical priors for learned image compression. *arXiv preprint arXiv:1809.02736*.

[16] STERN, M., SHAZEER, N. & USZKOREIT, J. 2018. Blockwise parallel decoding for deep autoregressive models. *arXiv preprint arXiv:1811.03115*.

[17] GREGOR, K., DANIHELKA, I., MNIH, A., BLUNDELL, C. & WIERSTRA, D. Deep autoregressive networks. International Conference on Machine Learning, 2014. PMLR, 1242-1250.

[18] LIM, B. & ZOHREN, S. 2021. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A,* 379**,** 20200209.

[19] ZHANG, J. 2017. Multivariate analysis and machine learning in cerebral palsy research. *Frontiers in neurology,* 8**,** 715.

[20] FAWAZ, H. I., FORESTIER, G., WEBER, J., IDOUMGHAR, L. & MULLER, P.-A. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery,* 33**,** 917-963.

[21] GERS, F. A., ECK, D. & SCHMIDHUBER, J. 2002. Applying LSTM to time series predictable through time-window approaches. *Neural Nets WIRN Vietri-01.* Springer.

[22] RAHMAN, S. A. & ADJEROH, D. A. 2019. Deep learning using convolutional LSTM estimates biological age from physical activity. *Scientific reports,* 9**,** 1-15.

[23] ANEJA, J., DESHPANDE, A. & SCHWING, A. G. Convolutional image captioning. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018. 5561-5570.

[24] LEA, C., VIDAL, R., REITER, A. & HAGER, G. D. Temporal convolutional networks: A unified approach to action segmentation. European Conference on Computer Vision, 2016. Springer, 47-54.

[25] XINGJIAN, S., CHEN, Z., WANG, H., YEUNG, D.-Y., WONG, W.-K. & WOO, W.-C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, 2015. 802-810.

[26] JI, S., XU, W., YANG, M. & YU, K. 2012. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence,* 35**,** 221-231.

[27] ZHANG, S., ZHOU, L., CHEN, X., ZHANG, L., LI, L. & LI, M. 2020. Network-wide traffic speed forecasting: 3D convolutional neural network with ensemble empirical mode

decomposition. *Computer-Aided Civil and Infrastructure Engineering,* 35**,** 1132-1147.

[28] HALLAC, D., VARE, S., BOYD, S. & LESKOVEC, J. Toeplitz inverse covariance-based clustering of multivariate time series data. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017. 215-223.

[29] CHAWLA, N. V., BOWYER, K. W., HALL, L. O. & KEGELMEYER, W. P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research,* 16**,** 321-357.

[30] ABADI, M., BARHAM, P., CHEN, J., CHEN, Z., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., IRVING, G. & ISARD, M. Tensorflow: A system for large-scale machine learning. 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), 2016. 265-283.

[31] COHEN, J. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement,* 20**,** 37-46.

Tekna

NFOGM
NORWEGIAN SOCIETY FOR
OIL AND GAS MEASUREMENT