INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 20, Issue: 3, Month: June, Year: 2025
Article Number: 7046, https://doi.org/10.15837/ijccc.2025.3.7046

CCC Publications

Development of SMOTET-LSTM Model Based on Hyperparameter Tuning for Fault Classification in Multi-Sensor Nodes

U. Şansal, M. Ersoy

Uğur Şansal*
Institute of Science
Department of Computer Engineering, Suleyman Demirel University
Süleyman Demirel University East Campus Çünür 32200 Isparta, Turkey
*Corresponding author: ugursansal@gazi.edu.tr

Mevlüt Ersoy
Department of Computer Engineering
Faculty of Engineering and Natural Sciences, Suleyman Demirel University
Süleyman Demirel University West Campus Çünür 32200 Isparta, Turkey
mevlutersoy@sdu.edu.tr

Abstract

In the context of Internet of Things (IoT) structures, sensor nodes have been observed to generate erroneous data due to their constrained operational capacity and position. The presence of faulty nodes can lead to significant challenges in communication, data traffic, and data evaluation. Consequently, it is imperative to segregate data obtained from faulty nodes from standard data. Concurrently, the identification of the specific fault type is paramount. The present study utilised machine learning and deep learning techniques to classify fault types, with the data collected from 54 sensors in a closed building over a period of 3 months. Initially, the performance analysis of the LSTM model was compared with that of Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Random Forest (RF) machine learning algorithms, given the utilisation of large amounts of data. Subsequently, as certain classes were characterised by limited data, data augmentation was implemented using synthetic data, and the SMOTET-LSTM model was developed through HPO (Hyper Parameter Optimization). This model demonstrated superior performance in comparison to the other algorithms.

Keywords: Fault Detection, Classification, Deep Learning, Machine Learning.

1 Introduction

A Wireless Sensor Network (WSN) can be defined as a structure in which sensors communicate with each other through nodes [1]. Sensors connected to these nodes collect data independently of each other in indoor and outdoor environments. The sensors affixed to these nodes have the capacity to collect data in an autonomous manner, both within indoor environments and in outdoor settings. Sensors enable the data in the environment to be converted into analog or digital signals.

These data can be temperature, humidity, sound, gas or particles, etc. Sensors are used by being integrated into an embedded system in WSN. The functionality of embedded systems is predicated on the capacity for data from sensors to be transferred between nodes using end-to-end wireless protocols [2]. Embedded systems are defined as devices that operate on low-power, with constrained computing power, memory, batteries, and storage capacity. There are many studies with WSNs in the fields of military, commercial, health and agriculture [3][4]. Cisco predicted that by 2023, the number of networked devices will be more than three times the population [5]. This will present difficulties in network management, given the volume of traffic on the Internet. In order to manage such large traffic, management mechanisms need to be developed on many objects. Because of the limited resources of IoT nodes and problems with access to the locations where they are placed, there are many problems with communication and data transfer. These problems can lead to increased network traffic, irregular battery and power usage, and incorrect data transfer in the IoT network. In addition, the instant data collection of the sensors increases energy consumption as it will continuously transmit the data to the network. This causes IoT nodes to shut down in a short time. Shutting down nodes require that the routing be changed to transfer data through the neighbour node. This situation causes blockages or interruptions in the communication of IoT nodes. For these reasons, investigation or detection of failure occurrence and taking measures will increase the service quality of the network. Error detection and intuition techniques have become interesting in IoT to provide more efficient traffic routing, efficient energy use, error-free data transmissions and accurate evaluations [6–10]. Fault detection techniques are those which are employed to locate faulty nodes within wireless sensor networks. These failures can be classified as permanent, intermittent and temporary [11]. Permanent faulty nodes produce faults continuously over time or do not respond to other nodes in the communication range [12]. Intermittent faulty nodes are usually prone to permanent failure [13]. Transient faulty nodes are not easily detected and become a probable event over time. Repeated tests may be required to detect transient faults. However, iterative fault detection algorithms increase the energy consumption and memory usage of the WSN node [14]. Due to the limited features of IoT nodes, this balance should be taken into account when performing effective fault detection. WSNs use different techniques such as mathematical, statistical, heuristic, and machine learning algorithms for fault detection [8,15–18]. The most important factor in detecting faults is the time intervals of repeated trials and the classification of the fault. Since running repeated fault detection algorithms will increase the energy load, it is necessary to determine the optimal time intervals. Determining the optimal time intervals is a situation that will reduce the limited energy consumption in Wireless Sensor Networks (WSNs). Fault detection systems represent a major field of study within the engineering discipline. Especially with the development of big data technologies, researchers have enabled the development of intelligent fault detection systems. Fault detection has often been considered as a classification problem. Support vector machine (SVM) model, one of the most widely used machine learning models as a classification model, is used in fault detection systems. Recent studies have witnessed significant advancements in the realm of fault diagnosis, largely attributable to the incorporation of artificial neural networks, fuzzy theory, machine learning algorithms utilising random forests, genetic algorithms (GA), the combination of GA and SVM, particle swarm optimization (PSO) and SVM.In the context of these recent developments, a noteworthy advancement is the introduction of fuzzy logic as a potential alternative to the limitations encountered by conventional fault detection computational methods[19]. The development of intelligent fault classification methodologies employing artificial neural networks (ANN) has been an active area of research in recent years [20]. (ANN) have been shown to possess certain advantages, including their ability to operate with incomplete information and to produce results with a high degree of efficiency, even when utilising minimal computational resources. However, it should be noted that they are not well-suited to datasets comprising a limited number of samples. In a separate study [21], The proposition of a technically sophisticated system for the identification of faults was presented, employing machine learning algorithms, with particular emphasis on the Random Forest algorithm. The efficacy of this approach is contingent on the quality of the training data, the employed learning algorithms, and the selected parameters. In [22], a genetic algorithm (GA) is proposed for fault detection. The GA is a very powerful computation that supports multi-objective functions and is very powerful for finding optimum solutions, although it is quite difficult to find the optimum solution due to its limitations, being time consuming and not suitable for problems with large variables. In the work cited in [23], a combination of a genetic algorithm (GA) and a support vector machine (SVM) has been proposed to form a hybrid system. This system is very useful for search and optimization, but it is unsuitable for large and uncertain data sets, which complicates parameter selection and leads to overfitting during the SVM process. As reported in this paper, a methodology integrating particle swarm optimization (PSO) and support vector machine (SVM) is proposed [24]. It has a strong generalisation capability for a high dimensional and nonlinear classification problem. However, this methodology is not without its drawbacks, chief among which is the sensitivity to local minima as well as the inevitable reliance on judicious selection of parameters for classification performance. Most of the data generated by IoT structures are considered as big data. Therefore, the LSTM classification model, which is a deep learning model, is also used in the problem of fault detection. In solving these problems, the data generated by sensors is usually considered as input data. The classification model is created according to predefined error values or reference value ranges [25–30]. In the study on artificial data generation [31], novel data on data generation facilitated the observation of new data, examples of minor classes separated from safe and borderline elements, and datasets exhibiting variable dimensions and instability characteristics. The preprocessing methods and classifier settings that formed the subject of the research have been extended. The study specifically examined the distributions of safe, borderline, rare, and outlier instances (corresponding to easy, medium, and difficult datasets), the classifiers employed, and the preprocessing methods utilized.Following the classification of errors in the data obtained from the sensors, it is possible that certain classes may exhibit a highly imbalanced distribution. In [32], the SMOTE algorithm is employed to address this imbalance, with the training and test ratio being equalised at 50%-50%, given the substantial volume of data.Integration of SMOTEBoost and bootstrap with minority data equalisation to other classes using SMOTE has been demonstrated in [33] to yield enhanced classification accuracy in the test dataset. The primary factor contributing to the efficacy of fault detection is the time interval between repeated trials, as well as the classification of errors. It is imperative to ascertain the optimal time intervals for executing repeated error detection algorithms, as their implementation will inevitably result in an augmented energy load. Determining the optimum time intervals is a situation that will reduce the limited energy consumption in IoTs

2 Material and Methods

2.1 Building Area Network System Model

Smart building systems consist of a combination of home area network and cloud computing systems. The Building area network shown in the figure includes four main components: smart devices, home gateway, VPN and cloud service provider.

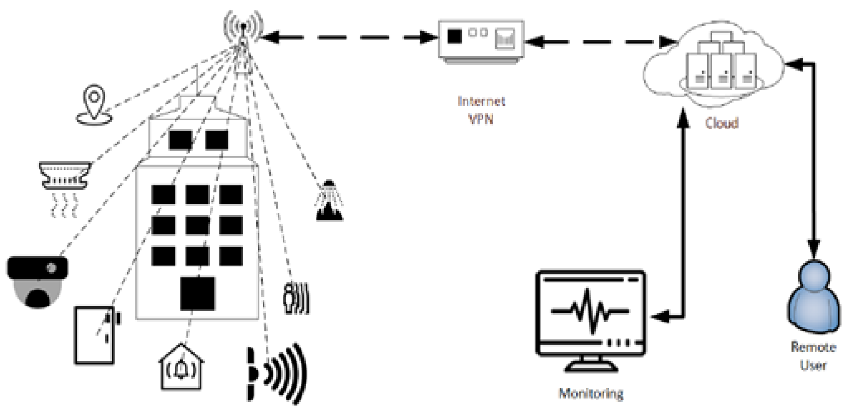

Figure 1: Typical Building Area Network[34]

2.2 Faults Taxonomy in WSN

WSNs are classified according to the time interval and location of the fault. It is classified as transient, intermittent, and permanent failures depending on the timing. Failures that occur for a limited duration are designated as transient failures. These failures can occur due to blockages in the network in nanoseconds and milliseconds, changing weather conditions. These failures disappear after a certain period of time. Intermittent failures, similar to transient failures, show that nodes have changed at some time intervals while they show normal behaviour. These periods are expected to be longer in intermittent failures. These faults are usually caused by incorrect readings of sensors. These errors are difficult to detect. Persistent errors are errors that do not disappear until the bug fix is resolved. The occurrence of such faults is ordinarily attributable to malfunctions in a local component of the WSN. [35]. The categorisation of the fault can be ascertained through the identification of its location. The fault itself is then classified as data- or system-centred. In the instance of data-centric failures, the characteristics of sensor readings are taken into consideration. In system-centric faults, the features of the system components used in WSNs are considered. Faults that occur according to the location of the fault are also explained in Table 1 [18,36,37].

Table 1: Error Types and Descriptions in WSN

Data Centric Faults	Offset Fault	It identifies predictable deviations in sensor readings. It is usually due to the deterioration of sensor calibrations.
	Gain Fault	The rate at which data detected in sensor readings is different from the expected data is referred to.
	Stuck-at Fault	When there is no change in sensor readings or zero.
	Out of bounds	Occurs when sensor readings are out of normal reading trend
	Spike Fault	Errors occur ringing times if the rate of change is too high over the same time period.
	Data Loss Fault	In case there is missing data in sensor readings over a certain time.
System Centric Faults	Calibration Fault	Errors in WS are usually calibration errors. Errors in the calibration formulas of sensors. Offset errors and earnings errors occur.
	Battery Failure	Causes incorrect transmission of data sent between nodes.
	Hardware failure	These are very common errors because WSN nodes are often present in harsh geographic conditions. In these failures, the defective equipment usually needs to be replaced. It causes permanent failures.

The limitations of the sensor nodes cause heavy computational systems not to be supported. In addition, its deployment in dangerous areas such as closed areas, factories, forests, railways, hard-to-reach lands causes difficulties in the fault detection of WSN. For this reason, there are many factors that force the diagnosis of WSN. The accuracy of fault detection in WSN increases the accuracy rates in evaluating data. In detecting the faults described in Table 1, the differences between the normal reading and the malfunction should be clearly revealed. A clear distinction must be made between faults. Fault detections must be fast because sensor readings are valuable information. In solving these challenging problem solving scenarios, it reveals that deep learning techniques and machine learning models can be a suitable solution for evaluating errors in challenging sensor readings. Classification of faults can be achieved with the classification techniques of these models.

2.3 Learning Models for Fault Detection

Classifiers are used to classify newly obtained observations according to the specified categories using the data set obtained from an environment for educational purposes. There are many classification algorithms used in the Literature on fault detection. The algorithms under scrutiny here are the support vector machine (SVM), the random forest (RF) and the K-nearest neighbour (KNN) algorithm[38-40]. SVM are a feed-forward, supervised machine learning algorithm employed in data analyses for regression and classification. They were developed at the ATT Bell laboratory [41]. The objective of SVMs is to minimise the upper bound of generalisation errors, as well as reducing experimental measurement errors. Consequently, the weight coefficient obtained in the training phase demonstrates superior generalisation and prediction performance for test data that has not been previously encountered. The classification of natural functions is a challenging problem. To address this challenge, SVM employs a transformation of the training data into a higher dimensional space, utilising linear functions to map the data. Rf algorithm is a community learning algorithm. Community learning represents a technique that involves the integration of numerous machine learning algorithms, with the objective of generating precise predictions that surpass those of any individual model. RF is composed of the merger of multiple decision trees. It is often used in regression and classification problems [42]. This algorithm is also created based on a random subset of observed data, with each tree created in decision trees.

2.3.1 LSTM

LSTM is an artificial neural network technique used for deep learning of the Long Short -Term Memory network. LSTM transfers information from the previous model to the next model. Time series analysis is called the analysis used based on previous data to predict a time-based variable. In time series analysis, the previous situation can affect the next situation. However, if sudden changes occur, it may be necessary to determine that this is a different situation. For this reason, these time series analyses can be performed with LSTM techniques. LSTM is an RNN variant that can effectively address the issue of detecting sudden changes in the degradient and time series (gradient vanishing) that are destroyed by introducing a collection of memory units. The recurrent neural network model ensures that the continuity of this information is available in layers in the common network model. Each neural network creates a loop within itself. Thanks to this structure, a repetitive structure is created by ensuring the continuity of knowledge. Evidently, in every instance, it can be seen that the function and weights applied to the network remain constant. At the time of t, the memory unit consists of three doors that contain all historical records to date. These are; (the entrance gate), forget the door and the exit door. The input values assigned to these gates range from 0 to 1. The LSTM network structures very suitable for sensor data sets such as temperature, pressure, gas, movement, light intensity, humidity, which contain time size. In this data, a few records of the current situation and the past status are sufficient for network training.In LSTM, input data can be defined as a multivariate time $X = _{X1},_{X2} + ... + _{Xt}$ series. The aim is to determine the fault detection algorithm consists of two parts. Prediction and detection. The initial stage in constructing the LSTM model is to reach a decision regarding the information derived from the cell states. This decision is made by a sigmoid layer with a forgetting gate. Eq. (1) controls the cell states in the LSTM model by outputting a number between 0 and 1 for each number $C_(t-1)$ A value of 1 means 'keep completely', while a value of 0 means 'get rid of completely' $h_(t-1)x_t$

$$g_t = \vartheta(W_g.[h_(t-1), x_t] + b_g) \tag{1}$$

The subsequent phase involves the determination of the effects to be stored in the cell state. Initially, a sigmoid layer known as the 'Gate Layer' determines the value to be updated (see Eq.(2)). Subsequently, a tanh layer generates new candidate vectors A (Ct) to be appended to the state ((see Eq. 3)). The amalgamation of these two steps engenders an update from an old cell state, Ct-1, to a new cell state, Ct.

$$i_t = \vartheta(W_i.[h_(t-1), x_t] + b_i) \tag{2}$$

$$C_t = \vartheta(W_c.[h_(t1), x_t] + b_i) \tag{3}$$

According to Eq. (4), we bump into the old state with the proportion of forgetting states and add each status value to be updated to scale.$g_t i_t x C_t$

$$C_t = g_t x C_{t_1} + i_t x C_t \tag{4}$$

Finally, predictions are intensively computed based on the filtered cell states. It is the function of the sigmoid layer to determine the state of the cell, which is then output via Eq. (5). The filtered cell states are then processed through a tanh function, thereby transforming the values within the range -1 and 1. The output is the sigmoid pass multiplied by the tanh value.

$$O_t = \vartheta(W_o.[h_(t-1), x_t] + b_i) \tag{5}$$

$$h_t = O_t x tanh(C_t) \tag{6}$$

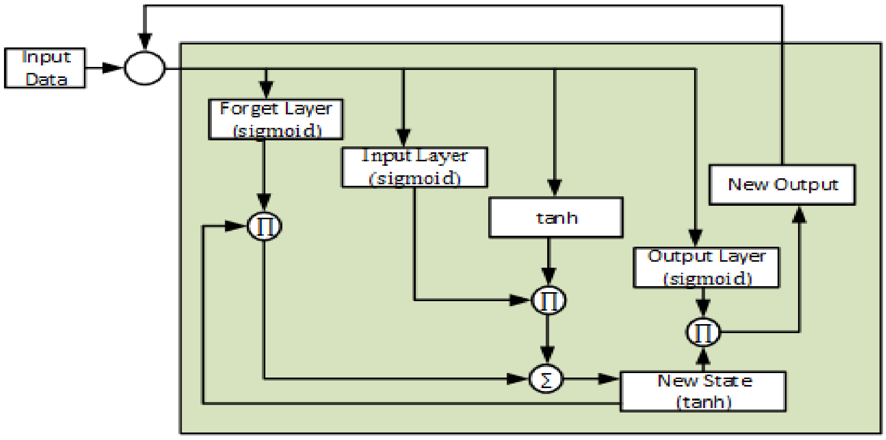

Figure 2: LSTM Typical Structure

Regarding this function, hyperbolic tangent is an appropriate choice for activation function for data limitation purposes: because being in range -1 and 1, the activation function supports both negative and positive values of standard incoming data.

2.3.2 KNN Algorithm

K-nearest neighbors (KNN) is a method proposed by Fix and Hodges that is easy to separate and has high separation between them. The KNN algorithm is predicated on the number of close neighbours, as determined by the calculation of the distance between new data and the sample data set. This is achieved by comparing the new data with the data in a particular sample set against other data [43]. The Data with unknown classes are monitored with other data in the training set and distances are tracked. According to the calculated distance, appropriate class assignment is made for new data [44]. First, it is aimed to reduce the size of the feature area.$A = (x_1, x_2, \ldots, x_m)$ and $B = (y_1, y_2, ..., y_m)$ by using the distance function.The most commonly used distance measures in calculating distance are the Euclidean, Minkowsky, Chebychev, Mahalanobis, and Cosine similarity distance in[45].

Table 2. Distance Metrics Used in KNN

Euclidean Distance	$\sqrt{\frac{\sum_{i=!}^{m} (x_i - y_i)^2}{m}}$	Minkowsky Distance	$\sqrt[p]{\sum_{i=1}^{m} \lvert x_i - y_i \rvert^p}$
Mahalanobis Distance	$\sqrt{(y_i - x_i)^T \; C^{-1} \; (y_i - x_i)}$	Chebychev Distance	$\lvert x_i - y_i \rvert$
Cosine similarity Distance		$\frac{\sum_{i=1}^{m} x_i y_i}{\sqrt{\sum_{i=1}^{m} x_i^2}\sqrt{\sum_{i=1}^{m} y_i^2}}$	

2.3.3 Support Vector Machine (SVM) Algorithms

The Support Vector Machine (SVM) algorithm is a computerised learning algorithm that is utilised for the purposes of classification or regression. The SVM algorithm is based on finding the maximum boundary between different data by moving the data to a high dimension where it can be linearly separated in the nonlinear sample space [46].

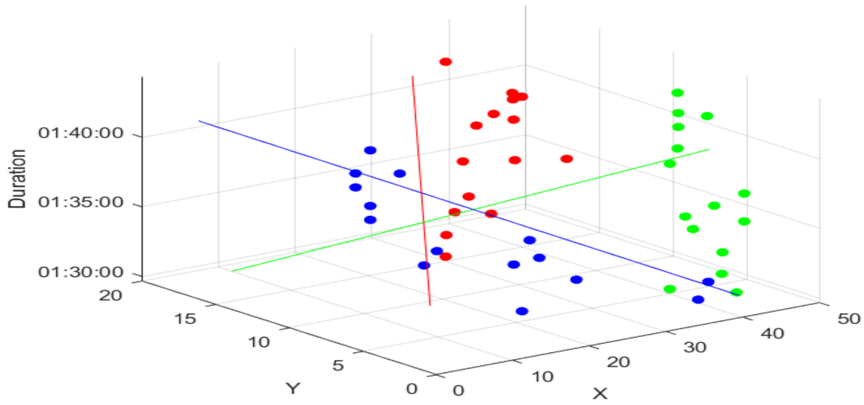

Figure 3: LSTM Typical Structure

The SVM algorithm is an optimally precise tool for the delineation of data into two categories, through the establishment of n-dimensional hyperplanes. In the field of support vector machines, problems are classified as either linearly or nonlinearly separable. The objective in cases of linearly separable problems is to ascertain the hyperplane that traverses the given features. In support vector machines, problems are categorised as linearly separable or non-linearly separable. In the case of linearly separable issues, the objective is to identify a hyperplane that intersects the features. The lines in which the properties of the classes are farthest from each other give the hyperplane. Figure 3 shows the support vectors and the hyperplanes. The linear lines passing through the middle of the plane acts as a separator for the classes.

2.3.4 Random Forest (RF) Algorithms

The random forest algorithm is a technique that has been demonstrated to be both effective and efficient when applied to a wide variety of problems. The most significant advantage of the random forest algorithm is its capacity to be utilised in classification problems with minimal parameter regression. In the context of the random forest algorithm, it is necessary to determine the optimum number of trees by gradually increasing the number of trees. In terms of the classification approach, to create a random forest algorithm, each node is divided into at least two branches until a decision is made to stop the node. Nodes are selected by a threshold so that data variance is minimal. By progressing through all these branches and nodes, a prediction is made somewhere [47].

2.4 Synthetic Minority Oversampling Technique

The synthetic data transfer process is undertaken with the utilisation of the following methods: SMOTE, SMOTETomek, OMC and UMC. SMOTE, an acronym for Synthetic Minority Oversampling Technique, is a method employed for the oversampling of minority classes in imbalanced data sets. There is a high imbalance in the classes of our sensor data obtained. SMOTETomek technology, the first persistence SMOTE process is used to multiply the minorities as in Figure 4. The second multiplication is removed using the Tomek Link technique to be distributed in the new data set used with multiplied minorities as in Figure 5. IFollowing the execution of the SMOTETomek operation, there will be a reduction in the number of majority class instances. This is due to the operation's capacity to identify and remove majority class instances that form Tomek links with minority class instances. The objective of this process is to enhance class separation by eliminating instances at the class boundary. UMC, Majority class reduction (Undersampling Majority Class) density class to minority class reduction method, the density class data is equalized to the minority class according to random or statistical samples. When the difference between the majority and minority classes is high, the data loss will be high, so it provides poor varieties. In the OMC Data Linkage Process, in the Oversampling Minority Class technique, new samples are added to the minority class until the minority class becomes equal to the density class to eliminate the imbalance of this oversampled manual dataset[48].

The basic equation for SMOTE can be written as [49].

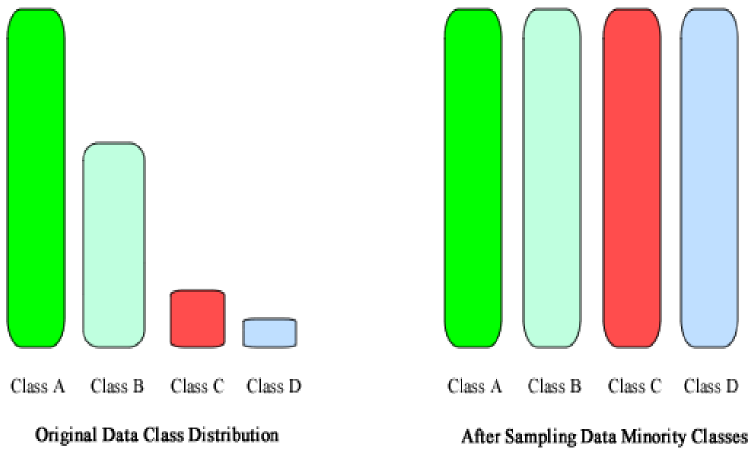

Figure 4: SMOTE Minority Class Balancing Process

$$X_{new} = X + (rand() * X_{nearest} - X)) \tag{7}$$

Where: X_{new} is the new synthetic sample generated by SMOTE, X is the minority sample that is being oversampled and $X_{nearest}$ is one of the nearest neighbors of X(based on some distance metric)[50] Eq. (7) creates a new sample between a randomly generated value (random minority sample (X) and one of its nearest neighbors ($X_{nearest}$). The number of minority samples is thus increased, and will help create a more balanced dataset for the model.

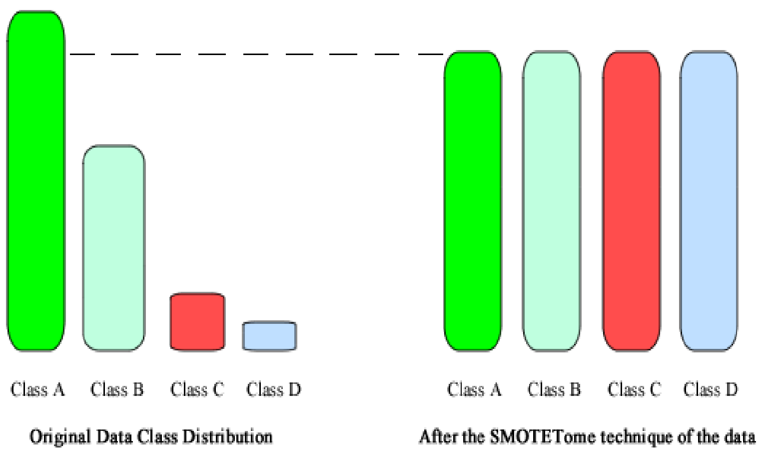

Figure 5: Minority Class Balancing Process with SMOTETomek

3 System Model

The data used in fault classification were obtained from sensor nodes placed in Intel Berkeley Research Lab [51]. Nodes are placed in positions determined on a two-dimensional plane as in Figure 6. Sensors placed in this closed area consist of 54 nodes that detect heat, light and humidity information. Data were obtained at half-hour intervals between February 2004 and April 2004.

It consists of 1048576 lines in total. Before entering the classification models, the data is first classified according to certain ranges. It is assumed that the temperature information obtained from indoor temperature sensors is in the range of 0-40 degrees Celsius, the light information is in the range of 0-10000 lux, and the humidity information is in the range of 0-100%.The fact that the data

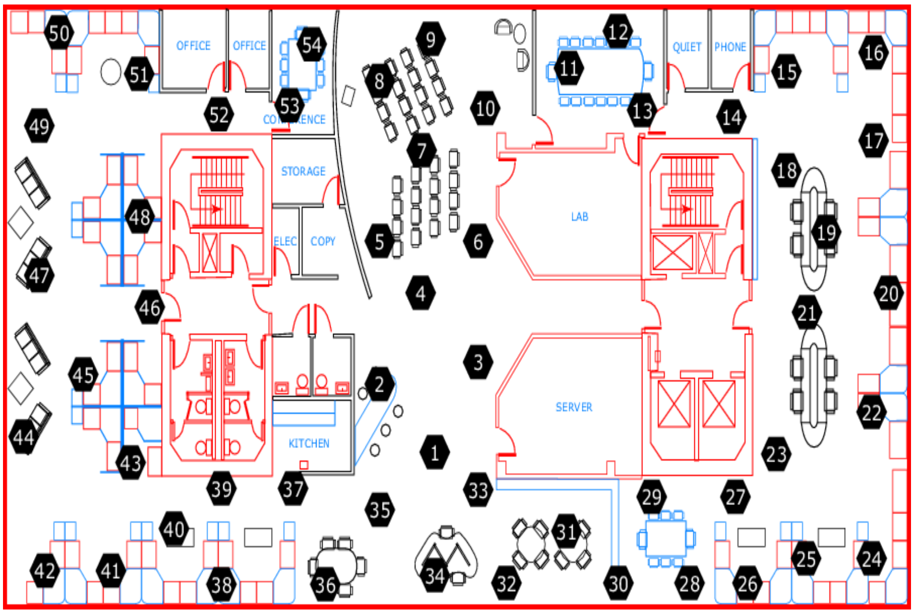

Figure 6: Indoor placement points for 54 sensors at Intel Berkeley Research Lab[51]

detected by the sensors are outside these ranges "offset fault"; reading the same data at intervals of change "gain fault"; Being 0 or not changing for a long time is defined as "stuck fault". In some of the perceived data, cases of having more than one error at the same time were encountered. In these cases, another row was created for the same data and different errors were specified. Table 3 shows an example data set.

Table 3: Example Data on The Sensor's Indoor Test Environment

date	time	Epoch	moteid	temparture	humidty	light	Fault Name
31.03.2004	03:38:16	2	1	122,15	-3,92	11,04	offsetFault
28.02.2004	00:59:16	3	1	19,99	37,09	45,08	normalData
28.02.2004	01:03:16	11	1	19,3	38,46	45,08	normalData
28.02.2004	01:06:16	17	1	19,17	38,8	45,08	normalData
28.02.2004	01:06:47	18	1	19,18	38,84	45,08	normalData
28.02.2004	01:08:46	22	1	19,15	38,94	45,08	normalData
...	...	...	...	...	...	...	...
21.03.2004	05:18:34	63880	10	19,97	47,57	0,92	offsetFault
21.03.2004	05:19:36	63882	10	19,97	47,47	0,92	offsetFault
21.03.2004	05:20:58	63885	10	19,98	47,54	0,92	offsetFault
21.03.2004	05:22:24	63888	10	19,97	47,54	0,92	offsetFault
21.03.2004	05:23:08	63889	10	19,96	47,54	0,92	offsetFault
21.03.2004	05:18:34	63880	10	19,97	47,57	0,92	offsetFault

Our aim is to detect errors in data read in real time as time series. In order to detect errors, data pre-processing, correlation analysis, development of prediction model and classification of errors were determined. Training was performed for 4 different classification values in prediction results: Offset Error, Gain Error, Compressed Error and normal data. Classification results obtained from SVM, KNN and RF algorithms used in data science related to error classification and LSTM were compared. Synthetic data generation process was performed to avoid imbalance in class data and generalizations in classification.

3.1 Hyperparameter Tuning Models

The efficacy of machine learning algorithms in training is contingent upon the optimisation of numerous internal hyperparameters [52]. While a hyperparameter may be known to have an effect on a model in general, the optimal tuning of a hyperparameter for a specific dataset remains unclear. Furthermore, many machine learning models possess a set of hyperparameters, which can interact in non-linear ways. Consequently, it is often necessary to record a hyperparameter setting that provides the best performance of a model on a given tuning data. This constraint requires a practical solution for optimizing hyperparameters.

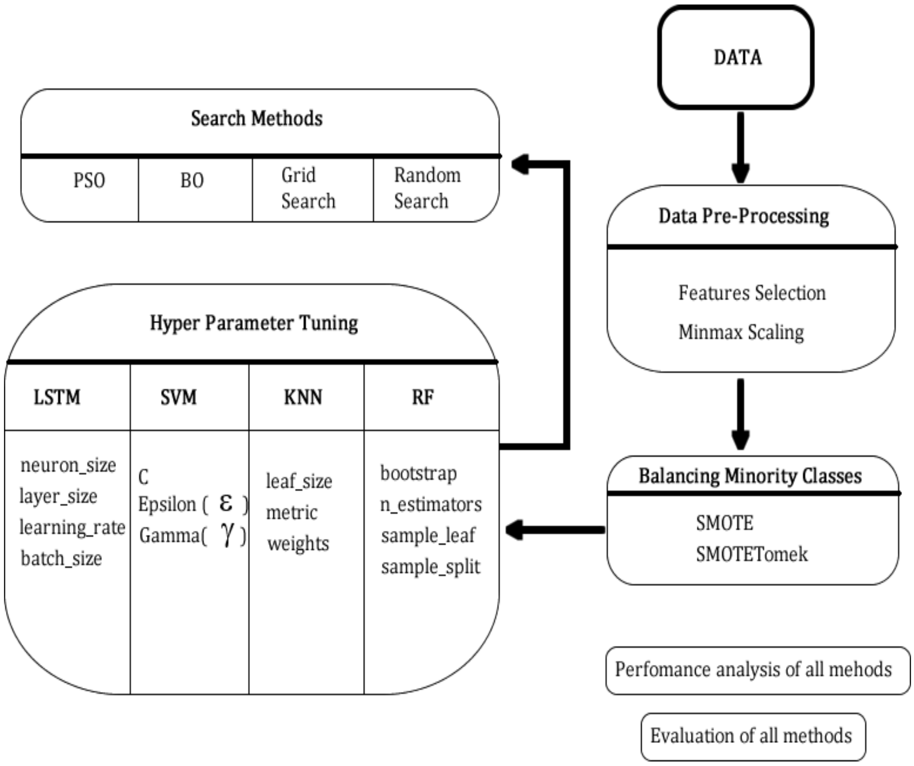

Figure 7: Schematic Of The Model Implemented For The Classification Of Sensor Data

Table 4: Intervals of Hyperparameter values according to models

Models	Parameters	Value Range
LSTM	Activation Function	sigmoid,tanh,relu,softmax
	Layer size	[5,10]
	Learning rate	[0.99,0.8]
	Batch size	32, 64, 128, 256
SVM	C	[1,10000]
	Epsilon	[0.001,0.1]
	Gamma	[0.00001-0.9]
KNN	n__neighbors	[5, 7, 9, 11, 13, 15,17]
	Metric	['manhattan','euclidean','minkowsk','Mahalanobis']
	weight	['uniform','distance']
RF	max__depth	[10, 20, 30, 40, 50]
	min__samples__leaf	[1, 2, 4,8]
	min__samples__split	[2, 10, 20]
	n__estimators	[100, 200, 400, 800]

In the current study, the following search algorithms are utilised for hyperparameter optimization

(HPO): particle swarm optimization (PSO), Bayesian optimization (BO), grid search and random search. It has been established that these algorithms are frequently employed to identify optimal solutions within the designated parameter ranges of machine learning (ML) and deep learning (DL) algorithms. The Bayesian search model is an optimization model founded upon Bayesian theory. This model constitutes a strategy that is employed to identify the minimum and maximum values of objective functions that are challenging to evaluate. It requires more computational power since a probabilistic model (e.g. Gaussian Process) must be updated at each iteration. Particle swarm optimization (PSO) can operate on continuous or discrete solution spaces. The correct tuning of the parameters significantly affects the success of the algorithm. It is generally faster because it is based on simple update rules. The extant literature suggests that Support Vector Machines (SVMs) with multiple classification features are considered to be an algorithm that can be used in the classification of errors. However, the parameters that must be determined for SVMs may present a challenge when setting parameters in a heterogeneous network. Consequently, it is necessary to ascertain the most suitable parameters for SVMs in a Building Area Network where heterogeneous reading operations are performed.

Table 5: Abbreviations used and their meanings

Abbreviations	Meaning
SVM	Support Vektor Machine
KNN	K Nearest Neighbors
RF	Random Forest
LSTM	Long Short Term Memory
BO	Bayesian Optimization
PSO	Particle Swarm Optimization
RS	Random Search
GS	Grid Search
SMOTE	SMOTE
SMOTET	SMOTETomek
OMC	Oversample Minority Class
UMC	Undersample Majority Class
HPO	Hiper Parameter Optimization

4 Result

The simulation results section consists of comparative analysis of the results obtained from SVM, KNN, RF, LSTM, SMOTE-LSTM and SMOTET-LSTM models. In order to achieve this objective, the performance of the algorithms was evaluated according to the mean absolute error (MAE), root mean square error (RMSE), correctly classified samples and training model generation time criteria.

These evaluations were conducted on a computer equipped with an i7 processor, 16 GB of RAM and a 640-core GPU, utilising the Python 3.7 compiler. Perusal of the scatter plot in Figure 6 will allow the data utilised in this study to be viewed.According to this plot, the data correctly detected by the sensor nodes is shown as “normalData”.It is seen that ‘normalData’ is more frequent than other data.In WSN, gainFault and stuckFault occur less frequently than offsetFault. LSTM model was used as a classification method in deep learning of the training data. It was compared with SVM, KNN and RF machine learning models to analyze the data and determine its accuracy. RMSE, MAE and MSE performance values showing the classification success for each model are shown in Table 6.

As demonstrated in Table 6, LSTM has been shown to outperform SVM, KNN and RF in accurately representing data variation. However, some classes were found to be generalised according to the metrics in the classification report of LSTM, SVM, KNN and RF.When the classification report

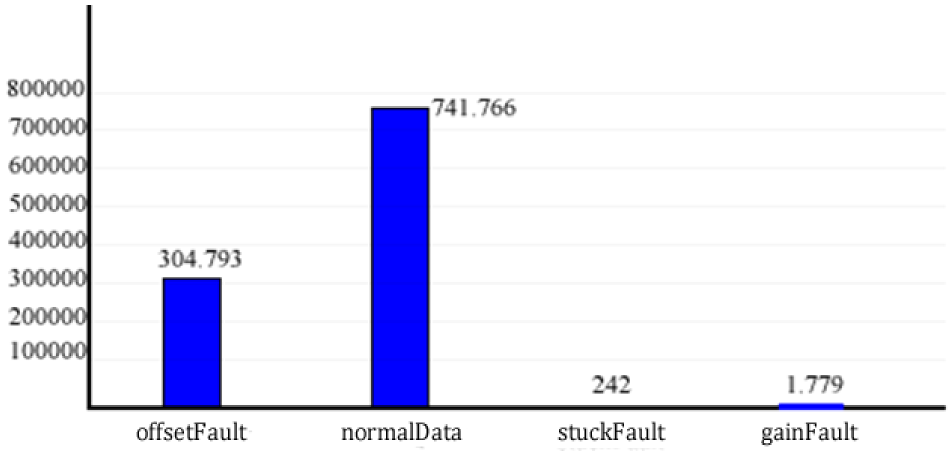

Figure 8: Distribution of Classes

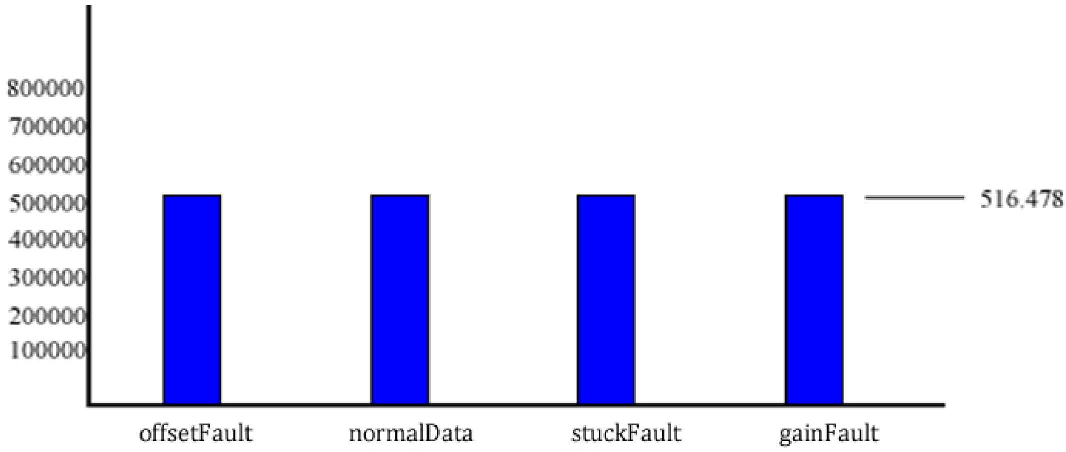

Figure 9: After Minority Class Balancing Process (SMOTET)

Table 6: Performance of the Classification Models

Model	HPO Tecnique	RMSE	MAE	Correctly Classified Examples
SVM	RS	0.2089	0.0941	%89.32
KNN	RS	0.2272	0.0563	%89.06
RF	RS	0.2503	0.0775	%96.80
LSTM	RS	0.2022	0.0373	%99.18
SMOTE-LSTM	BO	0.3337	0.1150	%99.05
SMOTET-LSTM	**BO**	**0.1736**	**0.0070**	**%99.36**

was analysed, it was observed that some classes were generalised.To overcome this imbalance, synthetic data were generated using SMOTE and SMOTETomek operations. The findings of this study demonstrate that the generalisation problem was effectively addressed in all four classes (offsetFault, NormalData, stuckFault, gainFault).As illustrated in Figure 10, minority classes are instantiated in other classes.The implementation of the SMOTE operation resulted in enhanced precision, recall and F1-score values, as depicted in Figure 11. Furthermore, the incorporation of SMOTETomek resulted in a more pronounced sampling of other classes (see Figure 12) and consequently enhanced performance.Following the experimental phase, where the RMSE and MAE values also attained the optimal value in Table 6, it was observed that the SMOTET-LSTM model achieved the maximum precision, recall and F1-score values.

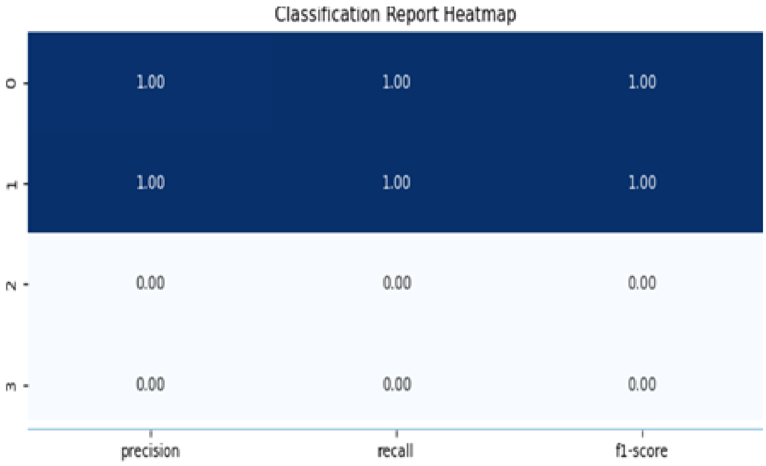

Figure 10: LSTM

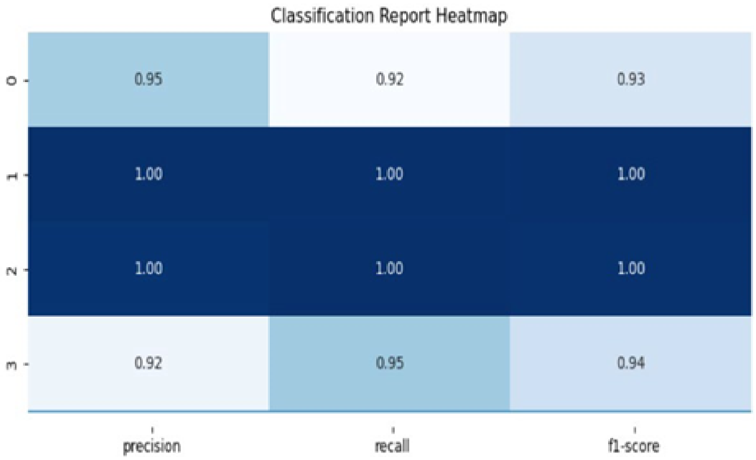

Figure 11: SMOTE-LSTM

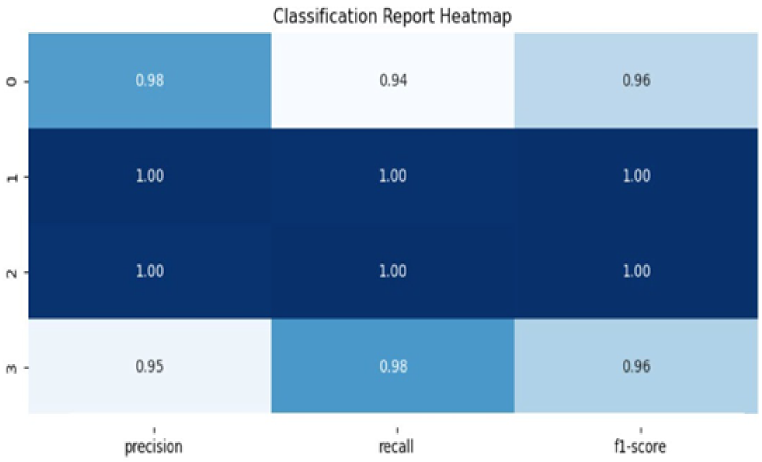

Figure 12: SMOTET-LSTM

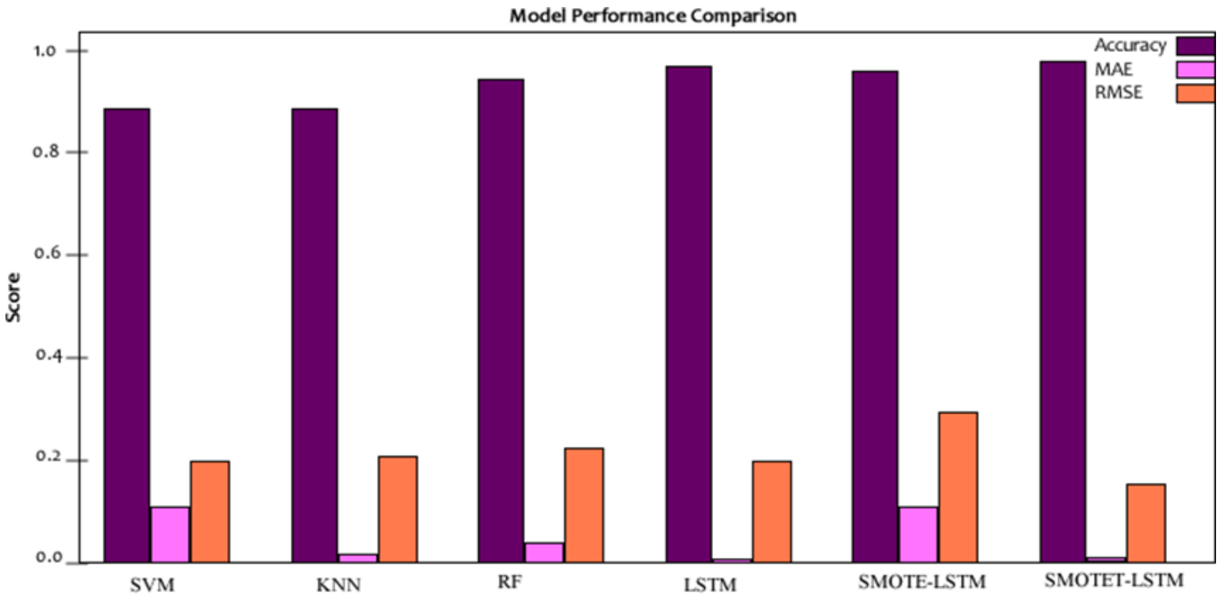

Figure 13: Comparison of Models

Table 7: Comparison Between Current Algorithms in Literature and the Proposed Algorithm

Algorithm	**Accuracy**
Prasojo et.al. [9]	97.62 %
Manoj et al. [10]	97.62 %
Prasojo et al. [11]	96.35 %
Zhang et al. [13]	96.50 %
Samah et al.[50]	98.22 %
SMOTET-LSTM	99.36 %

5 Conclusion

In many studies, classification models such as SVM, KNN and RF are used for error classification. The findings of our study show that these models perform optimally. However, in scenarios involving the use of significant data volumes in the model training process, the LSTM classification model is observed to be the fastest. LSTM, a deep learning technique, has been reported to achieve superior results in the context of classes characterised by large datasets and scarcity of data types. In this study, stuckFault and gainFault errors were generalised and accepted as normal data. To avoid this, synthetic data was generated and when tested with SMOTE, the RMSE and MAE results increased, although it gave good results compared to previous models. However, it was then tested with the SMOTETomek technique. The results of this model evaluation revealed that the RMSE and MAE reached the optimum performance, while the sensitivity, recall and f1-score values showed a significant improvement following this process.The Hyper Parameter Optimization (HPO) procedure has emerged as a very important element that gives different results between various values and models.The optimal result was obtained by adjusting the basic parameters. It was observed that SMOTET-LSTM gives favourable results when the dataset size increases. In real-time studies, a long time should not be allocated for the training time of the model. For this, it is critical to set up the training model quickly. Because the data is instantaneous and in case of error, it should be classified instantly by the system. In this case, it will be appropriate to perform the training of the data and the classification process with the cloud server over the end units and will also alleviate the load on the cloud server. Our results show that it is quite suitable for a WSN that can be used in real time. This research significantly advances the field of intelligent fault detection in sensor networks by effectively integrating sophisticated methodologies such as LSTM, SMOTETomek oversampling and hyperparameter optimization.

References

[1] Emperuman,M.;Chandrasekaran,S.(2020); Hybrid continuous density hmm-based ensemble neural networks for sensor fault detection and classification in wireless sensor network, *Sensors*, DOI:https://doi.org/10.3390/s20030745, 20,2020.

[2] Vuran,M.C.; Akan,Ö.B.; Akyildiz,I.F.(2004); Spatio-temporal correlation: Theory and applications for wireless sensor networks, *Comput. Networks*, DOI:https://doi.org/10.1016/j.comnet.2004.03.007,45,245—259,2004

[3] Yick.J.;Mukherjee,B.;Ghosal,D.(2008); Wireless sensor network survey, *Comput. Networks.*, 52, 2292—2330,2008, DOI:https://doi.org/10.1016/j.comnet.2008.04.002

[4] Ndiaye,M.; Hancke,G.P.; Abu-Mahfouz,A.M.(2017); Software defined networking for improved wireless sensor network management: A survey, *Sensors*,17 1—32,2017 DOI:https://doi.org/10.3390/s17051031

[5] Cisco,T.;Internet,A.(2020); 2020 CISO Benchmark Report *Comput. Fraud Secur*,2020,4. DOI:https://doi.org/10.1016/s1361–3723(20)30026-9.

[6] Chen,Y.; Son,S.H.(2005); A fault tolerant topology control in wireless sensor networks, *3rd ACS/IEEE Int. Conf. Comput. Syst. Appl*,269-–276,2005 DOI:https://doi.org/10.1109/AICCSA.2005.1387051.

[7] YU,M.; MOKHTAR,H.; MERABTI,.M.(2007); Fault Management in Wireless Sensor Networks, *IEEE Wirel. Commun*,13-–19.(2007)

[8] Mahapatro,A.; Panda,A.K.(2014); Choice of detection parameters on fault detection in wireless sensor networks: A multiobjective optimization approach, *Wirel. Pers. Commun*, 78, 649-–669,(2014) DOI:https://doi.org/10.1007/s11277-014-1776-1.

[9] Panda,M.; Khilar,P.M.(2014); Energy efficient distributed fault identification algorithm in wireless sensor networks, *J. Comput. Networks Commun*,(2014). DOI:https://doi.org/10.1155/2014/323754.

[10] Munir,A.; Antoon,J.; Gordon-Ross,A.(2015); Modeling and analysis of fault detection and fault tolerance in wireless sensor networks, *ACM Trans. Embed. Comput. Syst*,14, (2015). DOI:https://doi.org/10.1145/2680538.

[11] Swain,R.R.; Khilar, P.M.;Bhoi,S.K.(2018); Heterogeneous fault diagnosis for wireless sensor networks, *Ad Hoc Networks.*, 69, 15—37,(2018) DOI:https://doi.org/10.1016/j.adhoc.2017.10.012.

[12] Barooah,P.; Chenji,H.; Stoleru,R.; Kalmár-Nagy,T.(2012); Cut detection in wireless sensor networks, *IEEE Trans. Parallel Distrib. Syst* ,23, 483-–490, (2012) DOI:https://doi.org/10.1109/TPDS.2011.178.

[13] Sahoo,M.N.;Khilar,P.M.(2014); Diagnosis of wireless sensor networks in presence of permanent and intermittent faults, *Wirel. Pers. Commun.* ,78, 1571—1591,(2014) DOI:https://doi.org/10.1007/s11277-014-1836-6.

[14] Lee,M.H.; Choi,Y.H.(2008); Fault detection of wireless sensor networks, *Comput. Commun* ,31,3469—3475,(2008) DOI:https://doi.org/10.1016/j.comcom.2008.06.014.

[15] Chen,Q.; Lam,K.Y.; Fan,P.(2005); Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks, *IEEE Trans. Comput.* , 54, 1182—1183,(2005). DOI:https://doi.org/10.1109/TC.2005.140.

[16] Bagci,H.; Korpeoglu,I.; Yazici,A(2015); A distributed fault-tolerant topology control algorithm for heterogeneous wireless sensor networks, *IEEE Trans. Parallel Distrib. Syst.* , 26,914—923, (2015) DOI:https://doi.org/10.1109/TPDS.2014.2316142.

[17] Zidi,S.; Moulahi,T.; Alaya,B.(2017); Fault detection in wireless sensor networks through SVM classifier, *IEEE Sens. J.* ,18, 340-–347,(2018) DOI: https://doi.org/10.1109/JSEN.2017.2771226.

[18] Muhammed,T.; Shaikh,R.A(2017); An analysis of fault detection strategies in wireless sensor networks, *J. Netw. Comput. Appl.* , 78, 267—287, (2017) DOI: https://doi.org/10.1016/j.jnca.2016.10.019.

[19] Prasojo,R.A; Gumilang,H.; Maulidevi Suwarno,N.U.; Soedjarno,B.A.(2020); A fuzzy logic model for power transformer faults' severity determination based on gas level, gas rate, and dissolved gas analysis interpretation,Energies (Basel), 13 (4) (2020), p. 1009, 10.3390/en13041009

[20] Manoj,T.; Ranga,C.; Ghoneim,S.S.M.; Rao,U.M.; Abdelwahab,S.A.M.(2023); Alternate and effective dissolved gas interpretation to understand the transformer incipient faults, *IEEE Transact Dielectr Electr Insul* , 30,(3), pp. 1231–1239, (2023) DOI:10.1109/tdei.2023.3237795

[21] Prasojo,R.A.(2023); et al. Precise transformer fault diagnosis via random forest model enhanced by synthetic minority over-sampling technique, 220,Article 109361, (2023) *Electr Pow Systems Res*, DOI:10.1016/j.epsr.2023.109361

[22] Wu,Z.; Zhou,M.; Lin,Z.;Chen,X.; Huang,Y.(2021); Improved genetic algorithm and XG-BOOST classifier for power transformer fault diagnosis, *Front Energy Res*, 9,(2021) DOI:10.3389/fenrg.2021.745744

[23] Zhang,Y.; Li,X.; Zheng,H.; Yao,H.; Liu,J.;Zhang,C.; Jiao,J.(2019); A fault diagnosis model of power transformers based on dissolved gas analysis features selection and improved Krill Herd algorithm optimized support vector machine, *IEEE Access*, 7, pp. 102803-102811, (2019) DOI: 10.1109 /access. 2019.2927018

[24] Dashtdar,M.; Flah,A.; Hosseinimoghadam,S.M.S.; Reddy,C.R.; Kotb,H.;AboRas,K.M.; Bortoni,E.C. (2022); Improving the power quality of island microgrid with voltage and frequency control based on a hybrid genetic algorithm and PSO, *IEEE Access*,10, pp. 105352-105365,(2022) DOI:10.1109/ access.2022.3201819

[25] Jiahao,Y.; Jiang,X.; Wang,S.; Jiang,K.; Yu,X.(2020); SVM-BILSTM: A fault detection method for the gas station IoT system based on deep learning, *IEEE Access*,8,203712–203723,(2020), DOI: https://doi.org/10.1109/ACCESS.2020.3034939.

[26] Huang,H.; Zhao,L.; Huang,H.; Guo,S.(2020); Machine Fault Detection for Intelligent Self-Driving Networks, *IEEE Commun. Mag.*,58,40—46,(2020) DOI:https://doi.org/10.1109/MCOM.001.1900283.

[27] Bakhshi Kiadehi,K.; Rahmani,A.M.; Sabbagh Molahosseini,A.(2021); A fault-tolerant architecture for internet-of-things based on software-defined networks, *Telecommun. Syst*,77, 155—169,(2021) DOI: https://doi.org/10.1007/s11235-020-00750-1.

[28] Leu,F.Y.; Tsai,K.L.; Susanto,H.; Gu,C.Y.(2020); You,I.; A Fault Tolerant Mechanism for UE Authentication in 5G Networks, Mob. Networks Appl. (2020). DOI:https://doi.org/10.1007/s11036-019-01502-5.

[29] Qian,J.;Tiwari,P.; Gochhayat,S.P; Pandey,H.M.(2020); A noble double dictionary based ECG Compression Technique for IoTH, *IEEE Internet Things J.*,4662 1–1,(2020) DOI:https://doi.org/10.1109/jiot.2020.2974678.

[30] Saghiri,A.M.(2020); Blockchain Architecture, 161–176,(2020) DOI: https://doi.org/10.1007/978-981-13-8775-3_8.

[31] Wojciechowski, S.;Wilk,S.(2017); Difficulty Factors and Preprocessing in Imbalanced Data Sets: An Experimental Study on Artificial Data, *Foundations of Computing and Decision Sciences.*,vol. 42 no. 2, pp. 149-176. (2017) DOI:https://doi.org/10.1515/fcds-2017-0007

[32] Zhang, M.; Lu, J.; Ma, N.; Cheng, T.C.E.; Hua G. (2022); A Feature Engineering and Ensem-ble Learning Based Approach for Repeated Buyers Prediction, *International Journal of Computers Communications&Control*, 17(6), 4988, 2022. DOI:https://doi.org/10.15837/ijccc.2022.6.4988

[33] Cui,Y.;Ma.H.;Saha,T.(2014); "Improvement of power transformer insulation diagnosis using oil characteristics data preprocessed by SMOTEBoost technique," *in IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 21, no. 5, pp. 2363–2373,2014, DOI: 10.1109/TDEI.2014.004547.

[34] Verma,A.; Prakash,S.; Srivastava,V.; Kumar,A.; Mukhopadhyay,S.C.(2019); Sensing, Controlling, and IoT Infrastructure in Smart Building: A Review, *IEEE Sens. J.*,19, 9036—9046,(2019) DOI: https://doi.org/10.1109/JSEN.2019.2922409.

[35] Peng,N.; Zhang,W.; Ling,H.; Zhang,Y.; Zheng,L.(2018); Fault-tolerant anomaly detection method in wireless sensor networks, Inf. 9,(2018). DOI: https://doi.org/10.3390/info9090236.

[36] Kazmi,H.S.Z.; Javaid,N.; Awais,M.;Tahir,M.; Shim,S.O.; Bin Zikria,Y.(2019); Congestion avoidance and fault detection in WSNs using data science techniques, *Trans. Emerg. Telecommun. Technol.*, Trans. Emerg. Telecommun. Technol. 1–21,2019 DOI:https://doi.org/10.1002/ett.3756.

[37] Noshad,Z.; Javaid,N.; Saba,T.;Wadud,Z.; Saleem,M.Q.; Alzahrani,M.E.; Sheta,O.E.(2019); Fault detection in wireless sensor networks through the random forest classifier, *Sensors.*,19,1–21,2019. DOI: https://doi.org/10.3390/s19071568.

[38] Javaid,A.;Javaid,N.; Wadud,Z.; Saba,T.; Sheta,O.E.; Saleem,M.Q.; Alzahrani,M.E.(2019); Machine learning algorithms and fault detection for improved belief function based decision fusion in wireless sensor networks, *Sensors.*, 19,1—28,2019 DOI:https://doi.org/10.3390/s19061334.

[39] Jafarali Jassbi,S.; Moridi,E.(2019); Fault Tolerance and Energy Efficient Clustering Algorithm in Wireless Sensor Networks, *FTEC, Wirel. Pers. Commun.*, 107, 373—391,(2019) DOI:https://doi.org/10.1007/s11277-019-06281-6.

[40] Jia,S.; Ma,L.; Qin,D.(2018); Fault detection modelling and analysis in a wireless sensor network, J. Sensors. 2018. DOI: https://doi.org/10.1155/2018/7935802.

[41] Cristianini,N.; Ricci,E.(2008); Support Vector Machines, in: M.-Y. Kao (Ed.), Encycl. Algorithms, *Springer US*, Boston, MA, 2008: pp. 928—932. DOI: https://doi.org/10.1007/978-0-387-30162-4_415.

[42] BREIMAN,L.(2001); Random forests, Mach. Leatning. 45,5—32,(2001). DOI: https://doi.org/10.1007/978-3-662-56776-0_10.

[43] Masetic,Z.; Subasi,A.(2016); Congestive heart failure detection using random forest classifier., *Comput. Methods Programs Biomed*,130 , 54—64, (2016). DOI: https://doi.org/10.1016/j.cmpb.2016.03.020.

[44] Dilki,G.; Deniz Başar,Ö.; Yazar,S.; Author,C.; Tarihi,G.(2020); işletmelerin iflas tahmininde k- en yakın komşu algoritması üzerinden uzaklık ölçütlerinin karşılaştırılması, İstanbul Ticaret Üniversitesi Fen Bilim. Derg. 19 224—233,(2020) DOI:http://dergipark.gov.tr/ticaretfbd.

[45] Yu,H.; Yu,J.(2005); A Modified SMO Algorithm for SVM Regression and Its Application in Quality Prediction of HP-LDPE, *Springer*,Berlin, Heidelberg, 2005. DOI: https://doi.org/https://doi.org/10.1007/11539087_79.

[46] Sekulić, A.; Kilibarda, M.; Heuvelink, G.B.M.; Nikolić, M.(2020); Bajat, B. Random Forest Spatial Interpolation. *Remote Sens. 2020*, 12, 1687. DOI: https://doi.org/10.3390/rs12101687

[47] Wang,Z.; Wu,C.; Zheng,K.; Niu,X.; Wang,X.(2019); "SMOTETomek-Based Resampling for Personality Recognition," *IEEE Access*, vol. 7, pp. 129678—129689, 2019, DOI:10.1109/ACCESS.2019.2940061.

[48] Shelke1,M.S.; Deshmukh2,P.R.; Shandilya,V.K.(2017); "A Review on Imbalanced Data Handling UsingUndersampling and Oversampling Technique,", *International Journal of Recent Trends in Engineering and Research*, vol. 3, no. 4, pp. 444-–449, 2017. DOI: 10.23883/ijrter.2017.3168.0uwxm.

[49] Tra,V.; Duong,B.P.; Kim,J.M.(2019); Improving diagnostic performance of a power transformer using an adaptive over-sampling method for imbalanced data, *IEEE Transact Dielectr Electr Insul*, 26 (4) s. 1325–1333, (2019). DOI:10.1109/tdei.2019.008034.

[50] Gamel,Samah A.; Ghoneim,Sherif S.M; Sultan,Yara A.(2024); Improving the accuracy of diagnostic predictions for power transformers using a hybrid approach combining SMOTE and DNN, *Computers and Electrical Engineering*, Volume 117, 2024, 109232, ISSN 0045-7906, DOI:https://doi.org/10.1016/j.compeleceng.2024.109232.

[51] [Online].S. Madden, Intel Lab Data, Available: http://db.csail.mit.edu/labdata/labdata.html (Accesed on 2 February 2025).

[52] Falkner,S.; Klein,A.; Hutter,F.(2018); BOHB: Robust and efficient hyperparameter optimization at scale, *ArXiv*,(2018).

Cite this paper as:

Şansal, U.; Ersoy, M. (2025). Development of SMOTET-LSTM Model Based on Hyperparameter Tuning for Fault Classification in Multi-Sensor Nodes, *International Journal of Computers Communications & Control*, 20(3), 7046, 2025.
https://doi.org/10.15837/ijccc.2025.3.7046