IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 October 2023, accepted 29 October 2023, date of publication 6 November 2023, date of current version 9 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3330135

RESEARCH ARTICLE

Autoencoder Application for Anomaly Detection in Power Consumption of Lighting Systems

TOMASZ ŚMIAŁKOWSKI AND ANDRZEJ CZYŻEWSKI, (Member, IEEE)
Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland

Corresponding author: Tomasz Śmiałkowski (tomasz.smialkowski@pg.edu.pl)

This work was supported by the Polish National Centre for Research and Development (NCBR) through the European Regional Development Fund titled "INFOLIGHT-Cloud-Based Lighting System for Smart Cities" under Grant POIR.04.01.04/2019.

ABSTRACT Detecting energy consumption anomalies is a popular topic of industrial research, but there is a noticeable lack of research reported in the literature on energy consumption anomalies for road lighting systems. However, there is a need for such research because the lighting system, a key element of the Smart City concept, creates new monitoring opportunities and challenges. This paper examines algorithms based on the deep learning method using the Autoencoder model with LSTM and 1D Convolutional networks for various configurations and training periods. The evaluation of the algorithms was carried out based on real data from an extensive lighting control system. A practical approach was proposed using real-time, unsupervised algorithms employing limited computing resources that can be implemented in industrial devices designed to control intelligent city lighting. An anomaly detection algorithm based on classic LSTM networks, single-layer and multi-layer, was used for comparison purposes. Error matrix calculus was used to assess the quality of the models. It was shown that based on the Autoencoder method, it is possible to construct an algorithm that correctly detects anomalies in power measurements of lighting systems, and it is possible to build a model so that the algorithm works correctly regardless of the season of the year.

INDEX TERMS Anomaly detection, autoencoder, machine learning, road lighting systems, smart city, smart meters.

I. INTRODUCTION

Lighting management systems for roads, parks, and other public places are vital to Smart City infrastructure. Using such systems makes it possible to reduce electricity consumption and optimize control. Commonly used LED lighting, apart from the greater efficiency of the light source, enables effective reduction of road lighting levels, which brings additional energy savings and facilitates the implementation of adaptive lighting. Based on the data obtained from the sensors, the system dynamically adjusts the light intensity of a lamp or a group of lamps to weather or road conditions.

In lighting systems, there is a range of undesirable phenomena, i.e., failures, that should be detected as anomalies. Examples of such failures are switching off a group of lamps during the night or switching on lamps during the day when the lamps are switched on or off according to the schedule and incorrect power reduction in the system with such a reduction. A specific type of failure is energy theft - connecting an unauthorized energy receiver to the lamp's power supply circuit. For this type of anomaly, there is a direct relationship between their occurrence and instantaneous power consumption.

One of the most important functions of resource monitoring in management systems is anomaly detection, which can be done in Smart City systems based on data collected from various sources, primarily from electricity meters [1], [2].

Detecting anomalies in energy consumption is a well-studied problem in terms of different types of consumers: residential [3], commercial buildings [4], [5], and industry [6]. It is fostered by the widespread installation of smart electricity meters (Smart Meters); for example, in Europe, the penetration rate of smart electricity meters has reached 56% by the end of 2022. Additional factors include the provision of available metering databases to test research models. Unfortunately, road lighting systems have not yet been the subject of such interest; to the authors' knowledge,

The associate editor coordinating the review of this manuscript and approving it for publication was Essam A. Rashed.

the available publications have not yet dealt with studying anomalies in energy consumption for such systems. However, such a need exists since lighting systems are increasingly equipped with Smart Meters that record real-time power grid parameters. Incorporating such systems into the Smart City infrastructure makes it possible to acquire data from smart energy meters in real time and use it to analyze, forecast, and manage energy consumption.

The practical approach to detecting anomalies as system failures imposes specific requirements. First, it is preferred to use ''online'' algorithms [7], that is, algorithms that operate in real-time, analyzing recorded data and making a decision in a finite amount of time, as opposed to ''offline'' algorithms that analyze the complete data set after a period of observation. Second, unsupervised anomaly detection is preferred, which does not require data annotation, as they are classified based solely on the intrinsic properties of the dataset. Third, the algorithms must assume limited resources used for data processing so that the algorithms can be implemented in industrial equipment operating under harsh conditions.

The data read from the energy meters are recorded at equal time intervals, which makes them naturally form discrete time series. At the same time, the current, voltage, active power, and power factor values are recorded. However, only active power records will be used for this paper so that the analysis will be on Univariate Time Series. An overview of methods for detecting anomalies in time series has been presented previously in the literature, for example, in [8].

The power consumption of the lighting system is strictly periodic, with a period of 24 hours, due to the system's primary purpose, namely to illuminate public places from dusk to dawn. The course of power consumption corresponds to a unipolar rectangular wave with a variable filling factor, as the length of day and night change throughout the year. Depending on the time of year, the nighttime and lighting time lengthens or shortens, and this cycle repeats every year. Such variability in power consumption makes us deal with a non-stationary time series. When choosing representation models, it is necessary to account for this variability in the model (Concept Drift) through, for example, by correcting hyperparameters [9].

There are many methods for detecting anomalies in time series: statistical methods such as ARIMA/SARIMA, machine learning methods such as K-Means Clustering, Isolation Forest, or methods using neural networks (Deep Learning), among others: Convolutional Neural Networks, Long Short Term Memory (LSTM), Autoencoder. In the paper [10], the authors analyzed statistical methods based on the SARIMA model and Deep Learning methods based on LSTM networks.

Detecting anomalies in energy consumption from smart meters is used in fraud detection systems (FDS) for advanced metering infrastructure. A comparison of different clustering algorithms, FCM (fuzzy C-means), K-means, and SOM (Self-Organizing Map), was carried out in [11]. The FCM clustering method can be supplemented with DWT-based feature extraction [12]. Other clustering methods in FDS systems, maximum information coefficient, and clustering technique by fast search and finding density peaks are presented in [13].

Using Smart Meters in road lighting systems creates new opportunities for automatic diagnosis of undesirable phenomena such as lamp failures, deviations from the schedule, or theft of energy from the power supply network. This solution fits into the concept of Smart Cities, where using an adaptive lighting system creates new challenges for the monitoring function.

This paper deals with a method using the Autoencoder (AE) network model. The primary function of such a network is to reproduce input data at the output based on a learned pattern. However, when data in the input significantly differs from the memorized pattern, the AE will not be able to reproduce it correctly, which may indicate the presence of anomalies in the data. Since sequential data is being analyzed, models based on recurrent networks were chosen: the spline and LSTM networks.

II. RELATED WORKS

As mentioned, there is a lack of articles studying anomalous energy consumption by lighting systems. A very similar topic is dealt with by the authors of [14], although their focus is on methods for detecting and classifying abnormal behavior in the communication network of a lighting system using supervised learning. Various types of autoencoders and learning schemes used for unsupervised detection of energy consumption anomalies are analyzed in [15]. An autoencoder model using the clustering (ensemble) method was proposed, and a comparison of autoencoders based on unidirectional (Fully connected feed-forward) and convolutional (1D convolutional) networks was carried out, presenting techniques for evaluating autoencoder performance. The convolutional autoencoder achieved the best performance. The paper [6] investigated using an autoencoder using an LSTM network to detect elevated energy consumption in industrial applications, making it possible to identify improperly controlled, improperly maintained, or obsolete subsystems. An autoencoder using multiple layers of LSTM (Stacked AE) was tested in [18] for short-term forecasting capabilities. The model was extended with an attention mechanism (attention mechanism) based on assigning appropriate weights to input functions. The model was compared with several AE architectures using real measurement data for better forecasting accuracy.

On the other hand, in [19], a bi-directional LSTM (BiLSTM) network was used in the autoencoder to achieve greater accuracy in anomaly detection since two unidirectional LSTM networks more effectively extract features from temporal data. The threshold selection problem in anomaly detection was analyzed in [21]. A nonparametric, dynamic, and unsupervised threshold selection method was presented. In [23], it was shown that an autoencoder based on LSTM network can detect anomalies in predictable, unpredictable,

periodic, aperiodic, and quasi-periodic time series. It is also shown that the autoencoder can detect anomalies in short (30 samples) and long (more than 500 samples) time series. The application of the transfer learning method for time series classification is presented in [27].

III. SETUP

Models based on a recurrent neural network of the LSTM type were chosen for the study because it was designed to handle sequential data. For this type of network, a basic model consisting of an LSTM decoder and encoder, hereafter referred to as a single-layer model, and a deep model with an additional hidden layer, also of the LSTM type, hereafter referred to as a two-layer model, were studied. Fig. 1 and Fig. 2 show the structure of the analyzed networks using LSTM. For comparison, calculations were also carried out for AE with a one-dimensional braided network - 1D Convolutional, whose network structure is shown in Fig. 3.

In addition, for comparison purposes, classical LSTM, single-layer, and multi-layer networks were used for the anomaly detection algorithm, the structure of which is shown in Fig. 4.

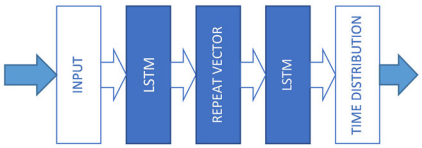

FIGURE 1. Single-layer autoencoder with LSTM network.

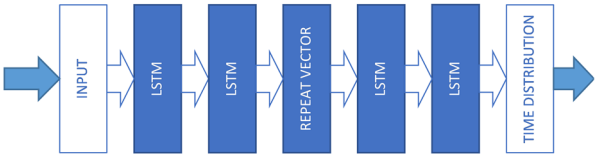

FIGURE 2. Dual-layer autoencoder with LSTM network.

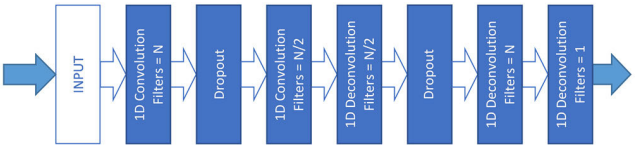

FIGURE 3. Autoencoder with 1D Convolution spliced network.

When constructing an autoencoder model, it is necessary to define several hyperparameters, the selection of which will determine both the correctness of the model and the computational complexity. The basic parameters of the selected networks and the designations used further to denote the simulations:

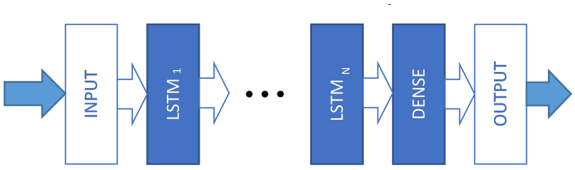

FIGURE 4. LSTM network applied to anomaly detection algorithm.

For AE LSTM single layer:

- AEL1U32 - Output space = 32
- AEL1U64 - Output space = 64
- AEL1U96 - Output space = 96
- AEL1U128 - Output space = 128

For AE LSTM two-layer:

- AEL2U32 - Output space 1st layer = 32, 2nd layer = 16
- AEL2U64 - Output space 1st layer = 64, 2nd layer = 32
- AEL2U96 - Output space 1st layer = 96, 2nd layer = 32
- AEL2U128 - Output space 1st layer = 128, 2nd layer = 32

For AE 1D Convolution:

- AEC7U32 - output space 1st layer = 32, 2nd layer = 16, convolution window = 7
- AEC7U64 - output space 1st layer = 64, 2nd layer = 32, convolution window = 7
- AEC9U64 - output space 1st layer = 64, 2nd layer = 32, convolution window = 9

For the LSTM network:

- LSTxUyy - x: number of layers = 1,2,3,4; yy: output space = 16, 32, 64

The research work described in this article was to analyze the effectiveness of detection algorithms based on different network architectures for selected sets of models. Since the anomaly detection algorithm is intended to be a real-time algorithm, it is essential to determine the minimum learning period for the models. Models requiring a large amount of training data, i.e., a long time, would have little practical use. Therefore, training periods of 3 to 7 days were chosen for the study.

Solving the anomaly detection problem can be approached as a binary classification method. Although the values of the analyzed signals have continuous values, defining a cutoff value will make the values classifiable as positive or negative, depending on whether the resulting value is higher or lower than the cutoff value.

The evaluation of anomaly detection algorithms is based on data from a lighting control system installed in a Polish city. The system controls over 4400 smart LED lamps and has 90 smart three-phase energy meters installed in control cabinets. Data from the meters is read at a 60-second interval and is transferred to a central database. Each record contains each phase's meter ID, timestamp, phase voltages, current values, active and apparent powers, and power factors. The database has recorded over 90 million records from June 2020 to December 2022. Fig. 5 shows an example of records for one of the meters. The representative database is available

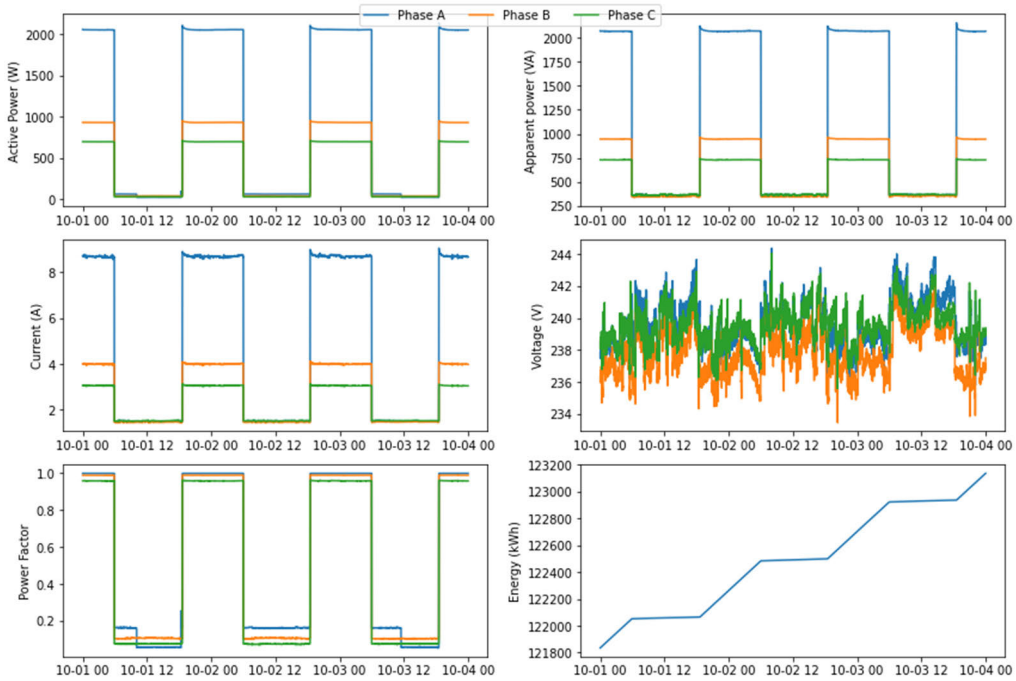

FIGURE 5. Example records of a smart energy meter.

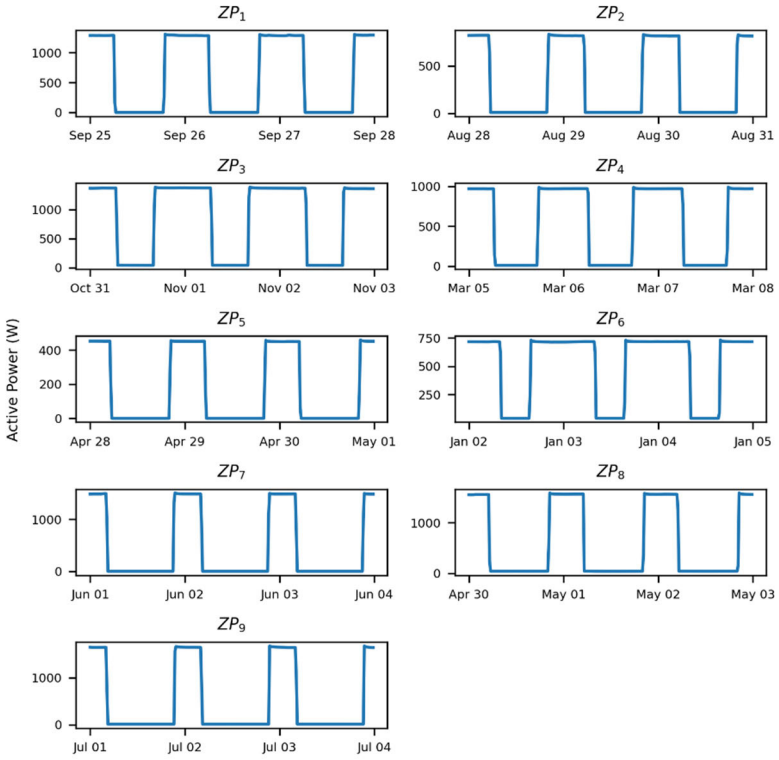

FIGURE 6. Example visualizations of power measurement data for ZP_1, ZP_2, ...ZP_9.

[28] and contains all the data for the calculations referred to in this article.

A 15-minute sampling period was used for the simulation, as this is the standard period for determining energy

consumption profiles for utility installations. Downsampling was used by calculating the average value over the period.

Simulations were performed on nine datasets labeled ZP_1, $ZP_2, \ldots ZP_9$. These data are from different meters for different seasons and contain varying values of power measurement amplitudes as they measure different circuits. Fig. 6 visualizes excerpts from these datasets..

All algorithms were implemented in Python version 3.10.5. The following libraries were used: pandas 1.3.5, NumPy 1.21.5, statsmodels 0.13.1, scikit-learn 1.0.2, TensorFlow 2.9.1, Keras 2.9.0. The calculations were carried out on a computer with an Intel® Core™ i7-7700HQ 2.8GHz processor equipped with 16GB RAM.

IV. ANALYSIS OF REGULAR DATA WITHOUT ANOMALIES

Since it was assumed that the autoencoder would be used for anomaly detection, it is important to answer how the constructed model reconstructs "correct" signals, i.e., without anomalies. The input data of the autoencoder are sequences of a fixed length corresponding to the periodicity of the signal under analysis. In the case of power measurements, the period equals 24 hours, with a sampling period of 15 minutes, giving 96 samples per sequence. A sequence of this length is generated in each step of the autoencoder operation. A measure of MAE - Mean Absolute Error (eq. 1), calculated for each pair of input and output sequences, is used to assess the correspondence of the reconstructed signal with the input signal:

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |y_t - y_t^P| \tag{1}$$

An example of the course of MAE values for training data is shown in Fig. 7.

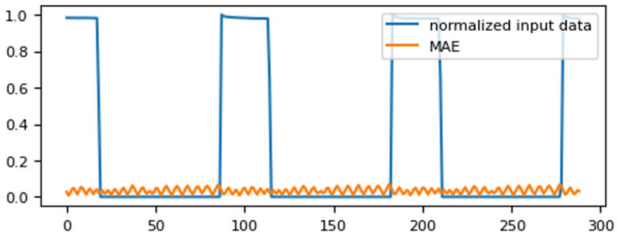

FIGURE 7. Example waveform of MAE value for autoencoder.

The input data undergoes min-max normalization before being processed by the AE algorithm to compare MAE values for different input data sets. The anomaly detection algorithm determines the maximum MAE value in the training period, which is the reference threshold for the MAE in the test period. It is assumed that the greater the consistency of the reconstructed signal, the greater the possibility of detecting anomalies because the detection threshold will be relatively low.

The simulation aims to detect the relationship between the autoencoder model configuration, the training period length, and the ability to reconstruct the input signal. An indicator was used: maximum MAE value - MAE_{max} to evaluate the reconstruction ability. This indicator's lowest possible value is expected for training and test data. However, these indicators are insufficient to assess the algorithm's potential. Since the detection threshold is the MAE_{max} value for the training data, the MAEmax value for the test data should be as close to it as possible. Therefore, an additional MR indicator calculated as the absolute value of the difference between the number 1 and the ratio of the MAE_{max} for the test period to the MAE_{max} for the training period is introduced (eq. 2):

$$MR = \left| \frac{Test\ MAE_{max}}{Train\ MAE_{max}} - 1 \right| \tag{2}$$

The MR value for regular signals, without anomalies, is expected to be close to 0.

For simulations based on the LSTM model, the way of calculating the MR index must differ because, in its case, a prediction of one value is made in one step for a given input sequence. Therefore, for LSTM, the measure for assessing the quality of reconstruction is the absolute error, denoted as AE (Absolute Error) (eq. 3):

$$AE_t = |y_t - y_t^P| \tag{3}$$

An example of the course of AE values for training data is shown in Fig. 8.

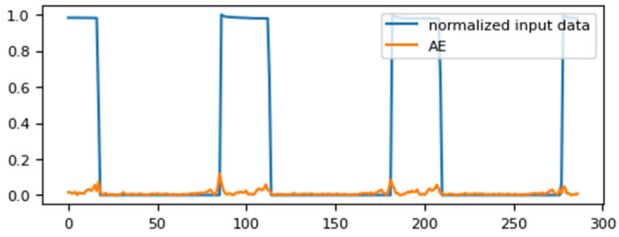

FIGURE 8. Example of AE value waveform for LSTM network.

For the LSTM model, the average values for a given period, or MAE, are used to calculate MR (eq. 4)

$$MR = \left| \frac{Test\ MAE}{Train\ MAE} - 1 \right| \tag{4}$$

An enumeration defined through Algorithm 1 was performed for each model.

An "early stopping" training method was used to train the models to avoid overtraining (overfitting). An MAE was adopted to monitor model performance, a validation dataset corresponding to 24 hours of measurements, a minimum index change of 0.0001, a number of epochs with no index change of 50, and a maximum number of epochs of 500. The remaining parameters are listed in Table 1.

The simulation results were then grouped for different lengths of the training period and different data sets, and the MR index was calculated. The average values of the MR index averaged over all data sets according to the length of

Algorithm 1 Calculate MAE_{max}

for train_period in set(3d..7d):
for data zp in set (ZP_1, ZP_2, ...ZP_9):

- create model
- train model
- make a prediction for train data
- calculate Train MAE for prediction
- make predictions for test data
- calculate MAE test for prediction
- calculate Train MAE_{max}, Test MAE_{max} and MR

TABLE 1. Autoencoder hyperparameters.

Hyperparameter	AE LSTM	AE 1D Convolution
Activation function	Rectified Linear Unit (ReLU)	Rectified Linear Unit (ReLU)
Loss function	Mean Squared Error (MSE)	Mean Squared Error (MSE)
Optimizer	Adaptive Moment Estimation (Adam)	Adaptive Moment Estimation (Adam)
Dropout	None	0.1
Batch Size	1	1

the training period are shown in Table 2 for autoencoder models and Table 3 for LSTM models. All tables starting from Table 2 are included in the Appendix. For AE models, the smallest value, i.e., the most favorable, was achieved by the simulation for the AEL2U32 model. For a training period of three days, simulations for spliced models achieved worse rates. For LSTM models, the lowest value was achieved by the simulation for the LST4U32 model and a training period equal to seven days. The indices of LSTM models are worse than those of AE models using LSTM. However, the latter is better than the indices for the AE weave and LSTM models.

Fig. 9 compares the average value of the MR index for selected models for those for which the average value of MR is the lowest.

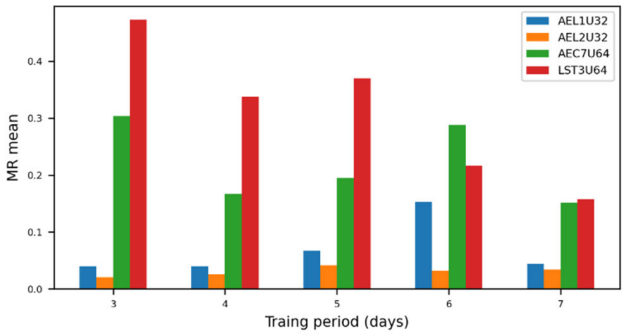

FIGURE 9. Comparison of the average value of the MR index for selected models.

Table 4 shows the standard deviation of the MR index for AE models and Table 5 for LSTM models. The simulation reached the smallest standard deviation value for the L2U32 model and the 3-day period. Fig. 10 shows a comparison of the standard deviation of the MR index for selected models.

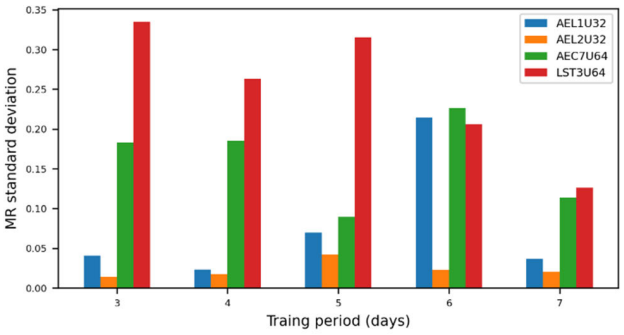

FIGURE 10. Comparison of the average value of the MR index for selected models.

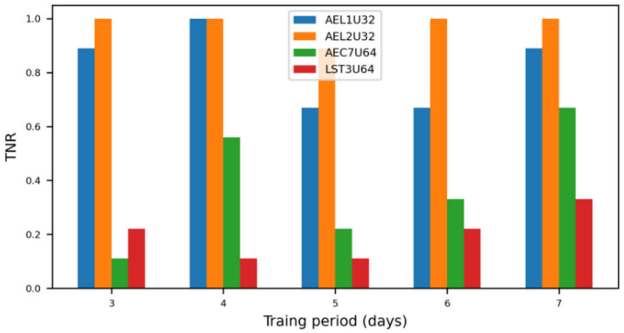

FIGURE 11. Comparison of True Negative Rate (TNR) values for selected models for Cutoff = 1.1 MAE_{max}.

Small values of the standard deviation of the MR index indicate that a given model correctly reconstructs data corresponding to situations without anomalies for all data pledges. According to the above results, the AEL2U32 model achieved both the smallest mean value of the MR index and the smallest standard deviation. Furthermore, the standard deviation for AE models using LSTM is lower than that of the spline AE models and LSTM models. It means AE models with LSTM are the most resistant to the concept drift phenomenon.

Confusion Matrix calculus was also used to assess the quality of the models. To be able to use binary classification, it is necessary to determine the threshold value. The natural choice is the MAE_{max} value for the training period; calculations were performed for a threshold 5% and 10% larger than the Train MAE_{max}. Error matrices were determined for all model/training period combinations for such thresholds.

Since regular signals without anomalies are analyzed at this stage, True Positive (TP) and False Negative (FN) values will always be zero. Therefore, the F1-score cannot be determined, but the True Negative Rate (TNR), also known as specificity, can be calculated. The calculated rates for different models and training periods are included in Table 6.

The calculated values indicate that none of the models classifies without error for a threshold equal to Train MAE_{max}.

In contrast, for a threshold greater than 10%, several AE models with LSTM can classify the data without error. Again, the AEL2U32 model shows the best signal reconstruction properties. Fig. 11 compares TNR values for selected models for Cutoff threshold = 1.1 MAE_{max}.

V. ANALYSIS OF DATA WITH ANOMALIES

The next step in verifying the effect of AE configuration on the performance of the detection algorithm is to conduct a test in which the reconstruction error for signals with anomalies is calculated. Since the detection algorithm must distinguish between correct and disturbed signals, a time series of measurements containing an anomaly is divided into three consecutive episodes. The first is the training period, the second is the episode without abnormality (T1), and the third is the episode with anomaly (T2), as shown in Fig. 12. Analogously, the calculations for data sets without anomaly ZP_1, ZP_2, …ZP_9, MAE_{max} and MR indicators for the T1 period are calculated. In addition, the AR index (eq. 5), which represents the amplitude of the anomaly, is calculated:

$$AR = \frac{T2MAE_{max}}{T1MAE_{max}} \tag{5}$$

Calculations were performed for six sets of measurements containing anomalies labeled $\mathbf{ZP_A, ZP_B, \ldots ZP_F}$. Fig. 13 visualizes excerpts from these datasets.

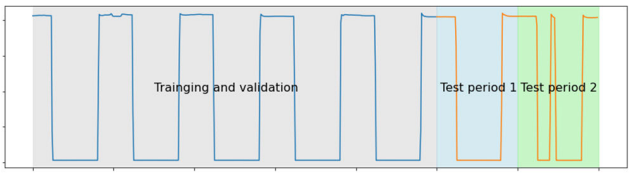

FIGURE 12. The principle of splitting a time series of measurements containing an anomaly.

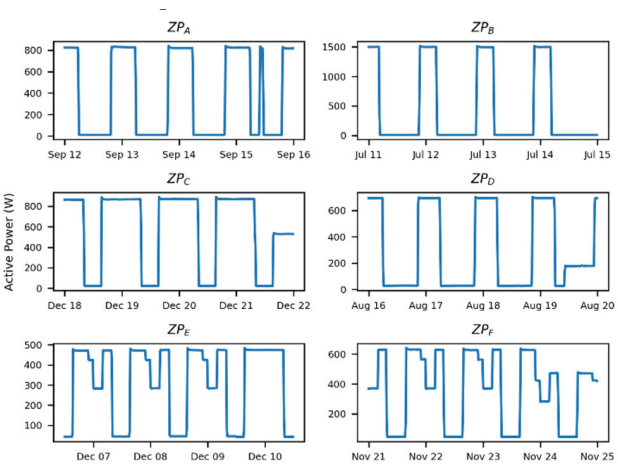

FIGURE 13. Example data with power measurements for $\mathbf{ZP_A, ZP_B, \ldots ZP_F}$.

ZP_A, ZP_B, ZP_C, ZP_D sets include abnormalities for control without reduction, ZP_E, ZP_F sets for control with reduction:

- $\mathbf{ZP_A}$ - switching on the lamps during the day
- $\mathbf{ZP_B}$ – no lamps on at night
- $\mathbf{ZP_C}$ – some of the lamps do not work – there was a decrease in power consumption
- $\mathbf{ZP_D}$ – switching on a group of lamps during the day or energy theft
- $\mathbf{ZP_E}$ – switching off the reduction schedule
- $\mathbf{ZP_F}$ - some of the lamps do not work - there has been a decrease in power consumption

A calculation defined by Algorithm 2 was performed for each model.

Algorithm 2 Calculate MAE_{max} for Train, T1, T2

for train_period in set (3d..7d):
  for data zp in set (ZP_A, ZP_B, …ZP_F):
- create model
- train model
- make a prediction for train data
- calculate Train MAE for prediction
- make a prediction for test period 1
- calculate T1 MAE for prediction
- make a prediction for test period 2
- calculate T2 MAE for prediction
- Calculate Train MAE_{max}, T1 MAE_{max}, MR, AR

The simulation results were grouped for the data sets, and the average value of MR and AR were calculated. The average values of the MR index are shown in Table 7 for the autoencoder models and in Table 8 for the LSTM models. Again, MR indices are significantly worse for LSTM models than AE models. This time, the AE weave models achieved comparable values to the AE LSTM models. Fig. 14 shows a comparison of the average value of the MR index for the T1 period for selected models.

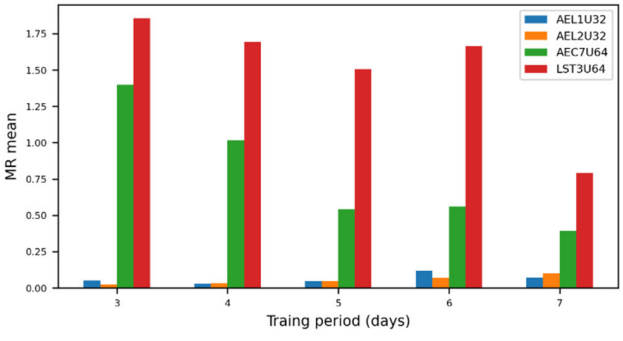

FIGURE 14. Comparison of the average value of the MR index for the T1 period for selected models.

The average values of the AR index are shown in Table 9 for autoencoder models and Table 10 for LSTM models. This time, the indices for LSTM models are better than those for AE models - the higher the AR value, the better. It is worth noting that the worst AR indicator was achieved by the AEL2U32 model, which outperformed the other models in

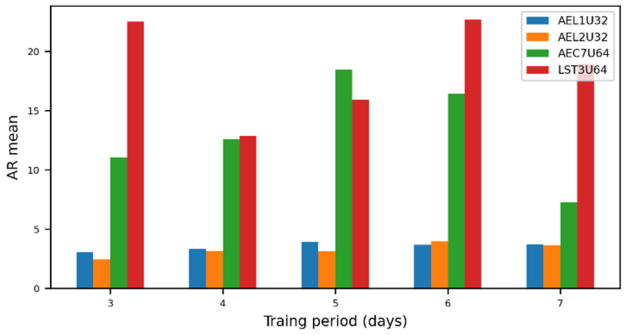

FIGURE 15. Comparison of the average value of the AR index for selected models.

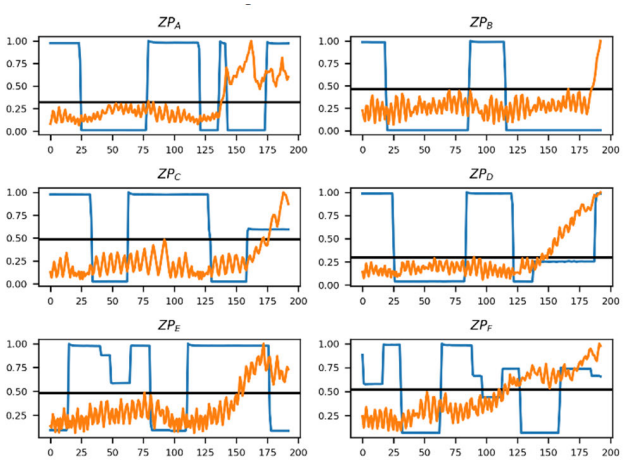

FIGURE 16. Course of MAE values during simulation for the AEL2U32 model and a training period equal to three days.

terms of other indicators. For all types of anomalies, the value of the AR index is sufficient for correct detection. Fig.15 compares the average value of the AR index for selected models.

Example waveforms of MAE values during simulation for AE LSTM models are shown in Fig. 16 and Fig. 17. Example waveforms of AE values during simulation for LSTM models are shown in Fig. 18.

Analogous to regular data analysis, error matrix calculus assessed the models' quality, and the same threshold values were used. This time we have a full set of values since the T1 period corresponds to a negative value and the T2 period corresponds to a positive value. If the signal in period T1 is below the threshold, it is a True Negative case, if above it is a False Positive case. If the signal in the period T2 is below the threshold, then it is False Negative, if above, then it is True Positive. Based on this relationship, F1-score values can be determined. The calculated indices for different models and training periods are included in Table 11. The calculated values show that for a threshold equal to Train MAE_{max} none of the models achieve F1-score values equal to 1. In contrast, for a threshold greater than 10%, several AE models with LSTM

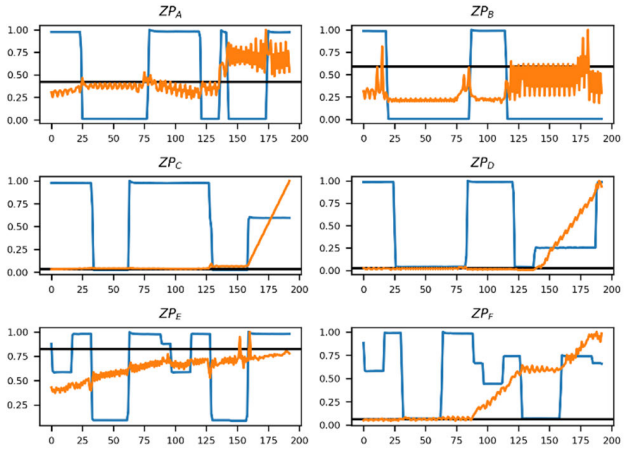

FIGURE 17. Course of MAE values during simulation for the AEC9U64 model and a training period equal to 3 days.

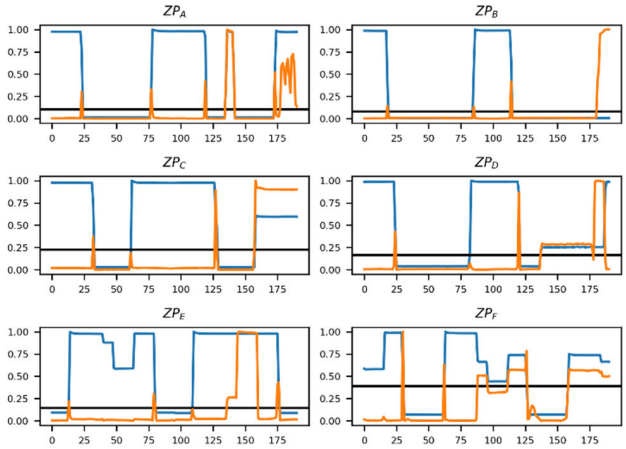

FIGURE 18. The course of AE values during simulation for the LST2U32 model and a training period equal to four days.

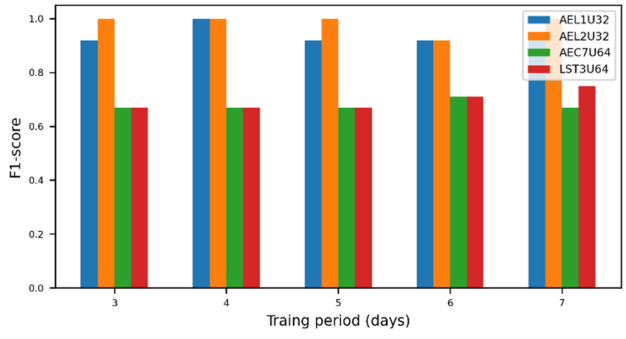

FIGURE 19. Comparison of F1-score values for selected models for Cutoff = 1.1 * MAE_{max}.

can achieve F1-score values equal to 1. Again, the AEL2U32 model shows the best signal reconstruction properties.

Additionally, the Area Under the Curve (AUC) and Mean Average Precision (MAP) were calculated for the

TABLE 2. The mean values of the autoencoder MR index depend on the training period's length. The seven smallest values are shown in bold, and the seven most significant values are in italics.

	MR mean for different training periods				
Model	**3**	**4**	**5**	**6**	**7**
AEL1U32	0.03972	0.03988	0.06690	0.15267	0.04380
AEL1U64	0.07119	0.08117	0.10567	0.09804	0.09911
AEL1U96	0.08823	0.06369	0.10800	0.10358	0.15012
AEL1U128	0.03922	0.09449	0.12566	0.16576	0.15726
AEL2U32	**0.02053**	**0.02553**	0.04144	**0.03200**	**0.03422**
AEL2U64	**0.03174**	**0.02733**	0.05909	0.05837	0.07681
AEL2U96	0.03566	0.05354	0.06531	0.11873	0.05653
AEL2U128	**0.02751**	0.07802	0.07531	0.05615	0.08491
AEC7U32	*0.39140*	0.22284	*0.27419*	0.15188	0.18893
AEC7U64	*0.30362*	0.16735	0.19461	*0.28809*	0.15175
AEC9U64	*0.43728*	*0.39176*	0.15731	0.16575	*0.27560*

TABLE 3. Mean values of the MR index for LSTM according to the length of the training period. The seven smallest values are shown in bold, and the seven largest values are in italics.

	MR mean for different training periods				
Model	**3**	**4**	**5**	**6**	**7**
LST1U16	0.48965	0.25068	0.43803	0.29908	0.38373
LST2U16	*0.49094*	0.32658	0.45108	0.32358	**0.19015**
LST3U16	*0.57576*	0.27486	0.44164	0.22017	0.23561
LST4U16	0.48872	**0.18265**	0.35482	0.32419	0.26679
LST1U32	*0.68413*	0.35629	0.42631	0.38450	0.28177
LST2U32	*0.55786*	0.48913	0.27319	0.32616	**0.18758**
LST3U32	*0.42473*	0.29295	0.45153	0.31814	0.22275
LST4U32	0.44101	0.32394	0.43125	**0.15984**	0.29952
LST1U64	*0.57964*	*0.50902*	0.43103	0.44499	0.21942
LST2U64	*0.56145*	0.37017	0.38252	**0.19887**	0.25431
LST3U64	0.47298	0.33755	0.37009	0.21653	**0.15768**
LST4U64	0.45209	0.42922	0.29879	**0.21309**	0.37715

TABLE 4. The standard deviation of the MR index for the autoencoder depends on the training period length. The seven smallest values are shown in bold, and the seven most significant values are in italics.

	MR standard deviation for different training periods				
Model	**3**	**4**	**5**	**6**	**7**
L1U32	0.04054	**0.02323**	0.06963	*0.21441*	0.03664
L1U64	0.06641	0.04982	0.11830	0.07804	0.13563
L1U96	0.06520	0.04581	0.05190	0.10848	*0.23210*
L1U128	0.03392	0.07674	0.12720	0.15669	0.07744
L2U32	**0.01413**	**0.01772**	0.04198	**0.02302**	**0.02032**
L2U64	0.02758	**0.01933**	0.03695	0.04999	0.08505
L2U96	0.03508	0.06703	0.05219	0.07404	0.03729
L2U128	**0.01915**	0.06084	0.04978	0.06018	0.07241
C2U32F7	*0.25389*	0.15225	*0.28044*	0.11570	0.14204
C2U64F7	0.18280	0.18554	0.08944	*0.22632*	0.11392
C2U64F9	*0.38679*	*0.24987*	0.09925	0.13623	0.20721

Autoencoder models. The results are included in Table 12. The advantage of models based on LSTM is again confirmed.

Based on the above results, it can be concluded that the best results in terms of anomaly detection were achieved by algorithms based on autoencoder with an LSTM network. Firstly, these algorithms achieved the best F1-score value and the best TNR index. Secondly, they reached the lowest MR values. Thirdly, they require the shortest training periods.

VI. CONCLUSION

The novelty of the article is the use of a deep learning method using an Autoencoder model to detect anomalies in road lighting systems by analyzing data from smart energy

TABLE 5. The standard deviation of the MR index for LSTM depends on the length of the training period. The seven smallest values are in bold, and the seven most significant ones are in italics.

	MR standard deviation for different training periods				
Model	**3**	**4**	**5**	**6**	**7**
LST1U16	0.29052	0.17758	0.34742	0.25714	0.20520
LST2U16	0.24895	0.21375	0.27734	*0.43845*	0.16660
LST3U16	*0.64618*	0.19682	0.22013	0.15577	0.14379
LST4U16	0.25356	0.17409	0.37181	*0.38517*	0.15508
LST1U32	*0.45274*	0.29368	0.22723	0.26143	**0.11404**
LST2U32	*0.56283*	0.26621	0.15845	0.25481	0.21186
LST3U32	*0.41948*	**0.11085**	0.29760	0.21608	**0.14232**
LST4U32	0.24379	0.24489	0.18590	**0.08492**	0.21785
LST1U64	*0.53526*	0.33809	0.20905	0.28912	0.14422
LST2U64	0.37516	0.27956	0.27083	0.16193	**0.09611**
LST3U64	0.33479	0.26347	0.31550	0.20626	**0.12649**
LST4U64	0.34028	0.38413	0.15491	**0.12602**	0.31797

TABLE 6. True Negative Rate (TNR) values for different models and training periods. Values equal to 1.0, indicating error-free classification, are shown in bold, and values equal to 0.89, corresponding to one misclassification, are shown in italics.

	True negative rate for different training periods														
	Cutoff = MAE_{max}					**Cutoff = 1.05 * MAE_{max}**					**Cutoff = 1.1 * MAE_{max}**				
Model	**3**	**4**	**5**	**6**	**7**	**3**	**4**	**5**	**6**	**7**	**3**	**4**	**5**	**6**	**7**
AEL1U32	0.33	0.44	0.56	0.33	0.44	0.78	*0.89*	0.67	0.56	0.67	*0.89*	**1.00**	0.67	0.67	*0.89*
AEL1U64	0.11	0.33	0.44	0.22	0.44	0.56	0.44	0.56	0.56	0.78	0.67	0.67	0.78	0.56	*0.89*
AEL1U96	0.44	0.22	0.44	0.22	0.22	0.56	0.56	0.44	0.56	0.44	0.78	0.78	0.67	0.56	0.67
AEL1U128	0.56	0.44	0.22	0.11	0.11	0.67	0.56	0.33	0.33	0.22	*0.89*	0.67	0.44	0.56	0.44
AEL2U32	0.11	0.44	0.44	0.33	0.22	*0.89*	**1.00**	*0.89*	*0.89*	0.78	**1.00**	**1.00**	*0.89*	**1.00**	**1.00**
AEL2U64	0.44	0.56	0.44	0.33	0.33	0.78	0.78	0.67	0.78	0.67	**1.00**	**1.00**	*0.89*	0.78	0.78
AEL2U96	0.78	0.44	0.56	0.44	0.22	*0.89*	0.67	0.67	0.56	0.56	**1.00**	*0.89*	*0.89*	0.56	*0.89*
AEL2U128	0.00	0.67	0.56	0.22	0.33	*0.89*	0.67	0.56	0.78	0.56	**1.00**	0.67	0.78	*0.89*	0.67
AEC7U32	0.11	0.00	0.11	0.33	0.67	0.11	0.11	0.22	0.56	0.78	0.22	0.22	0.44	0.56	0.78
AEC7U64	0.00	0.33	0.22	0.33	0.44	0.11	0.44	0.22	0.33	0.67	0.11	0.56	0.22	0.33	0.67
AEC9U64	0.00	0.00	0.11	0.56	0.56	0.00	0.00	0.22	0.56	0.56	0.11	0.11	0.44	0.67	0.56
LST1U16	0.11	0.00	0.11	0.11	0.00	0.11	0.00	0.22	0.33	0.22	0.11	0.22	0.22	0.33	0.22
LST1U32	0.00	0.00	0.11	0.00	0.11	0.00	0.11	0.11	0.00	0.22	0.00	0.22	0.11	0.33	0.44
LST1U64	0.11	0.11	0.00	0.11	0.44	0.33	0.22	0.00	0.22	0.44	0.33	0.22	0.00	0.33	0.44
LST2U16	0.00	0.22	0.11	0.56	0.56	0.00	0.44	0.22	0.56	0.56	0.00	0.56	0.33	0.56	0.56
LST2U32	0.11	0.00	0.11	0.22	0.22	0.11	0.11	0.11	0.22	0.22	0.11	0.22	0.11	0.22	0.22
LST2U64	0.00	0.00	0.11	0.22	0.44	0.22	0.00	0.11	0.22	0.56	0.33	0.00	0.22	0.22	0.56
LST3U16	0.33	0.00	0.00	0.11	0.56	0.33	0.00	0.00	0.11	0.56	0.33	0.00	0.11	0.33	0.56
LST3U32	0.11	0.11	0.00	0.11	0.44	0.11	0.11	0.00	0.22	0.44	0.11	0.22	0.00	0.33	0.44
LST3U64	0.11	0.00	0.11	0.22	0.11	0.22	0.11	0.11	0.22	0.22	0.22	0.11	0.11	0.22	0.33
LST4U16	0.11	0.00	0.11	0.22	0.33	0.22	0.00	0.11	0.44	0.33	0.22	0.00	0.11	0.44	0.33
LST4U32	0.00	0.11	0.00	0.22	0.44	0.11	0.33	0.22	0.44	0.56	0.22	0.33	0.22	0.56	0.67
LST4U64	0.22	0.11	0.22	0.33	0.44	0.22	0.22	0.33	0.33	0.44	0.33	0.22	0.33	0.44	0.44

TABLE 7. Mean values of the MR index for the T1 period for autoencoder models. The seven smallest values are in bold, and the seven most significant ones are in italics.

	MR mean for different training periods				
Model	**3**	**4**	**5**	**6**	**7**
AEL1U32	0.05113	**0.02915**	0.04854	0.11875	0.07110
AEL1U64	0.04808	0.04426	0.10781	0.05398	0.17951
AEL1U96	**0.03402**	0.10697	0.08853	0.21281	0.25065
AEL1U128	0.04834	0.09466	0.23444	0.17984	0.24968
AEL2U32	**0.02498**	**0.03160**	0.04816	0.07021	0.10062
AEL2U64	**0.04222**	0.07969	0.05141	0.10844	0.08041
AEL2U96	**0.03135**	0.06051	0.07953	0.10300	0.06142
AEL2U128	**0.02739**	0.08884	0.07176	0.08669	0.10745
AEC7U32	*0.83450*	*0.93544*	0.68598	0.59947	0.54827
AEC7U64	*1.39895*	*1.01770*	0.54210	0.55952	0.39343
AEC9U64	*0.93681*	*0.99969*	0.33812	*0.77687*	0.49274

TABLE 8. Mean values of the MR index for the T1 period for LSTM models. The seven smallest values are shown in bold, and the seven largest values are in italics.

	MR mean for different training periods				
Model	**3**	**4**	**5**	**6**	**7**
LST1U16	*1.95641*	1.26665	**0.88221**	1.52120	**0.91190**
LST2U16	1.87311	1.71382	1.17654	1.57622	0.96567
LST3U16	*2.09356*	1.91868	1.38721	1.35630	1.06257
LST4U16	1.52183	1.69565	1.81549	1.76192	**0.89908**
LST1U32	1.87048	1.14084	1.04336	1.22610	1.18020
LST2U32	1.48922	*1.93748*	1.60217	1.75640	**0.93197**
LST3U32	1.15805	1.63450	1.34003	1.36031	**0.78087**
LST4U32	1.48923	1.51632	1.45469	1.63770	1.30821
LST1U64	*2.01765*	1.09728	0.98244	1.44700	1.03788
LST2U64	*2.19303*	*2.06107*	1.63591	1.64868	**0.77278**
LST3U64	1.85623	1.69333	1.50508	1.66571	**0.79131**
LST4U64	1.44792	1.70508	*1.94945*	1.67431	1.10373

TABLE 9. Average values of the AR index for AE models. The seven most significant values are highlighted in bold, and the seven smallest values are in italics.

	AR mean for different training periods				
Model	**3**	**4**	**5**	**6**	**7**
AEL1U32	*3.05012*	*3.31733*	3.90280	3.68182	3.70649
AEL1U64	3.48337	4.08550	4.33475	5.44829	4.89508
AEL1U96	3.87193	4.88916	7.92353	8.21517	9.92131
AEL1U128	3.51464	4.87596	4.28970	6.63162	6.40066
AEL2U32	*2.44564*	*3.15305*	*3.12341*	3.97364	3.62779
AEL2U64	*2.80578*	4.32591	5.71755	7.18621	4.86811
AEL2U96	*2.89398*	4.33993	4.76396	5.49804	6.29359
AEL2U128	3.43460	4.87736	5.55112	6.26768	9.19711
AEC7U32	8.15378	11.70802	**13.26902**	10.08079	**18.67025**
AEC7U64	11.03933	12.60973	**18.46674**	**16.43032**	7.23564
AEC9U64	7.51256	11.44228	**16.86993**	**16.92520**	**16.23037**

TABLE 10. Average values of the AR index for LSTM models. The seven most significant values are highlighted in bold, and the seven smallest in italics.

	AR mean for different training periods				
Model	**3**	**4**	**5**	**6**	**7**
LST1U16	15.60151	18.62488	15.08596	20.87465	18.20012
LST2U16	21.34068	18.52103	**28.68775**	22.20669	20.19267
LST3U16	19.36079	17.16739	18.87715	**28.23481**	19.80395
LST4U16	19.50385	*12.37200*	16.37253	20.15143	18.81375
LST1U32	19.76128	17.79041	17.15843	17.74367	15.84883
LST2U32	15.03598	*13.97635*	15.73592	17.71605	18.59864
LST3U32	19.71739	*13.18574*	20.58818	**23.77156**	22.06210
LST4U32	22.48515	*11.60377*	**28.84765**	19.59814	19.99510
LST1U64	18.12847	19.39665	17.41461	21.13131	*14.49876*
LST2U64	20.37969	15.16072	16.75528	**25.24467**	22.21257
LST3U64	22.49238	*12.84961*	15.90037	**22.69671**	18.94497
LST4U64	16.03103	14.71438	*12.65234*	**24.79009**	15.35840

meters. Based on the Autoencoder method, it is possible to construct an algorithm that correctly detects anomalies in lighting systems' power measurements. Furthermore, it is possible to build the model so that the algorithm works correctly for the entire data recording period, regardless of the season. The training period of the model can be equal to three days, which means that the algorithm is ready to detect anomalies after this time. Simulations also showed the advantage of the autoencoder model over LSTM models.

Based on the results obtained so far, the authors plan to use other methods, such as transformer networks and clustering methods based on FCM. Further development of experimental research is planned, the expected result of which is automatic supervision of the operation of lighting systems carried out by artificial intelligence.

TABLE 11. F1-score values for different models and training periods. Values equal to 1.0, indicating error-free classification, are highlighted in bold, while values equal to 0.92, corresponding to one misclassification in italics.

	F1-score for different training periods														
	Cutoff = MAE_{max}					Cutoff = 1.05 * MAE_{max}					Cutoff = 1.1 * MAE_{max}				
Model	**3**	**4**	**5**	**6**	**7**	**3**	**4**	**5**	**6**	**7**	**3**	**4**	**5**	**6**	**7**
AEL1U32	0.75	0.71	0.75	0.75	0.80	0.86	***0.92***	0.86	0.80	***0.92***	***0.92***	**1.00**	***0.92***	***0.92***	***0.92***
AEL1U64	0.71	0.71	0.67	0.71	0.86	0.86	0.86	0.80	0.80	***0.92***	**1.00**	***0.92***	0.86	***0.92***	***0.92***
AEL1U96	0.86	0.80	0.75	0.75	0.75	**1.00**	0.86	0.86	0.80	0.75	**1.00**	***0.92***	***0.92***	0.80	0.80
AEL1U128	0.80	0.71	0.75	0.80	0.75	***0.92***	0.75	0.75	0.80	0.80	**0.92**	0.80	0.80	0.86	0.80
AEL2U32	0.80	0.75	***0.92***	0.75	***0.92***	**1.00**	**1.00**	**1.00**	***0.92***	**1.00**	**1.00**	**1.00**	**1.00**	***0.92***	**1.00**
AEL2U64	0.75	0.75	0.75	0.80	0.80	0.86	***0.92***	0.86	0.86	0.86	**1.00**	***0.92***	**1.00**	0.86	***0.92***
AEL2U96	0.86	0.80	0.80	0.86	**1.00**	***0.92***	0.86	0.80	***0.92***	**1.00**	**1.00**	***0.92***	**1.00**	***0.92***	**1.00**
AEL2U128	0.80	0.86	0.75	0.71	0.86	***0.92***	0.86	0.86	0.80	***0.92***	**1.00**	***0.92***	***0.92***	0.86	**1.00**
AEC7U32	0.67	0.67	0.71	0.71	0.75	0.67	0.67	0.71	0.71	0.71	0.67	0.67	0.62	0.62	0.77
AEC7U64	0.67	0.67	0.67	0.67	0.62	0.67	0.67	0.67	0.71	0.62	0.67	0.67	0.67	0.71	0.67
AEC9U64	0.67	0.67	0.67	0.71	0.75	0.67	0.67	0.67	0.71	0.75	0.67	0.67	0.67	0.71	0.80
LST1U16	0.71	0.71	0.67	0.67	0.67	0.71	0.71	0.67	0.71	0.71	0.71	0.71	0.71	0.75	0.71
LST1U32	0.67	0.71	0.75	0.67	0.71	0.67	0.71	0.75	0.67	0.75	0.67	0.71	0.75	0.67	0.75
LST1U64	0.67	0.67	0.71	0.71	0.71	0.71	0.67	0.71	0.75	0.71	0.71	0.67	0.71	0.75	0.71
LST2U16	0.67	0.67	0.67	0.67	0.75	0.71	0.67	0.67	0.67	0.75	0.71	0.67	0.67	0.67	0.80
LST2U32	0.71	0.67	0.67	0.71	0.67	0.75	0.67	0.67	0.71	0.67	0.75	0.67	0.71	0.71	0.67
LST2U64	0.71	0.67	0.67	0.67	0.71	0.71	0.67	0.67	0.67	0.71	0.71	0.67	0.67	0.67	0.71
LST3U16	0.67	0.67	0.75	0.71	0.75	0.67	0.67	0.75	0.71	0.75	0.67	0.67	0.75	0.75	0.80
LST3U32	0.71	0.67	0.75	0.67	0.75	0.71	0.67	0.75	0.67	0.75	0.71	0.67	0.75	0.67	0.75
LST3U64	0.75	0.67	0.71	0.67	0.67	0.75	0.67	0.71	0.67	0.67	0.75	0.67	0.71	0.67	0.67
LST4U16	0.67	0.67	0.67	0.71	0.67	0.67	0.67	0.67	0.71	0.67	0.67	0.67	0.67	0.71	0.67
LST4U32	0.67	0.67	0.67	0.67	0.75	0.67	0.67	0.67	0.67	0.75	0.67	0.67	0.67	0.71	0.75
LST4U64	0.67	0.67	0.67	0.67	0.75	0.67	0.67	0.67	0.67	0.75	0.67	0.67	0.67	0.67	0.75

TABLE 12. Area Under the Curve (AUC) and mean average precision (MAP) for autoencoder models for different training periods.

	AUC					MAP				
Model	**3**	**4**	**5**	**6**	**7**	**3**	**4**	**5**	**6**	**7**
AEL1U32	1.00	1.00	1.00	1.00	1.00	0.83	0.83	0.83	0.83	0.83
AEL1U64	1.00	1.00	1.00	1.00	1.00	0.83	0.83	0.83	0.83	0.83
AEL1U96	1.00	1.00	1.00	1.00	1.00	0.83	0.83	0.83	0.83	0.83
AEL1U128	1.00	1.00	1.00	1.00	1.00	0.83	0.83	0.83	0.83	0.83
AEL2U32	1.00	1.00	1.00	1.00	1.00	0.83	0.83	0.83	0.83	0.83
AEL2U64	1.00	1.00	1.00	1.00	1.00	0.83	0.83	0.83	0.83	0.83
AEL2U96	1.00	1.00	1.00	1.00	1.00	0.83	0.83	0.83	0.83	0.83
AEL2U128	1.00	1.00	1.00	1.00	1.00	0.83	0.83	0.83	0.83	0.83
AEC7U32	0.86	0.92	0.86	0.86	0.86	0.72	0.78	0.76	0.76	0.73
AEC7U64	0.81	0.94	0.83	0.86	0.83	0.71	0.79	0.71	0.73	0.75
AEC9U64	0.81	0.81	1.00	0.94	0.92	0.67	0.69	0.83	0.78	0.76

APPENDIX

DETAILED RESULTS OF EXPERIMENTS

See Tables 2–12.

REFERENCES

[1] M. Ali, P. Scandurra, F. Moretti, and L. Blaso, "Architecting a big data-driven software architecture for smart street lighting," in *Proc. IEEE 20th Int. Conf. Softw. Archit. Companion (ICSA-C)*, Mar. 2023, pp. 1–10.

[2] M. Mishra, J. Nayak, B. Naik, and A. Abraham, "Deep learning in electrical utility industry: A comprehensive review of a decade of research," *Eng. Appl. Artif. Intell.*, vol. 96, Nov. 2020, Art. no. 104000.

[3] S. Arastehfar, M. Matinkia, and M. R. Jabbarpour, "Short-term residential load forecasting using graph convolutional recurrent neural networks," *Eng. Appl. Artif. Intell.*, vol. 116, Nov. 2022, Art. no. 105358.

[4] Y. Himeur, K. Ghanem, A. Alsalemi, F. Bensaali, and A. Amira, "Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives," *Appl. Energy*, vol. 287, Apr. 2021, Art. no. 116601.

[5] T. Mendes, P. J. S. Cardoso, J. Monteiro, and J. Raposo, "Anomaly detection of consumption in hotel units: A case study comparing isolation forest and variational autoencoder algorithms," *Appl. Sci.*, vol. 13, no. 1, p. 314, Dec. 2022.

[6] C. Kaymakci, S. Wenninger, and A. Sauer, "Energy anomaly detection in industrial applications with long short-term memory-based autoencoders," *Proc. CIRP*, vol. 104, pp. 182–187, Jan. 2021.

[7] A. A. Cook, G. Misirli, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.

[8] M. Braei and S. Wagner, "Anomaly detection in univariate time-series: A survey on the state-of-the-art," 2020, *arXiv:2004.00433*.

[9] G. Fenza, M. Gallo, and V. Loia, "Drift-aware methodology for anomaly detection in smart grid," *IEEE Access*, vol. 7, pp. 9645–9657, 2019.

[10] T. Śmiałkowski and A. Czyżewski, "Detection of anomalies in the operation of a road lighting system based on data from smart electricity meters," *Energies*, vol. 15, no. 24, p. 9438, Dec. 2022.

[11] M. Zanetti, E. Jamhour, M. Pellenz, M. Penna, V. Zambenedetti, and I. Chueiri, "A tunable fraud detection system for advanced metering infrastructure using short-lived patterns," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 830–840, Jan. 2019.

[12] R. Qi, J. Zheng, Z. Luo, and Q. Li, "A novel unsupervised data-driven method for electricity theft detection in AMI using observer meters," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.
[13] K. Zheng, Q. Chen, Y. Wang, C. Kang, and Q. Xia, "A novel combined data-driven approach for electricity theft detection," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1809–1819, Mar. 2019.
[14] T. Andrysiak and Ł. Saganowski, "Anomaly detection for smart lighting infrastructure with the use of time series analysis," *J. Universal Comput. Sci.*, vol. 26, no. 4, pp. 508–527, Apr. 2020.
[15] C. Fan, F. Xiao, Y. Zhao, and J. Wang, "Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data," *Appl. Energy*, vol. 211, pp. 1123–1135, Feb. 2018.
[16] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019.
[17] C. Chahla, H. Snoussi, L. Merghem, and M. Esseghir, "A novel approach for anomaly detection in power consumption data," in *Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, 2019, pp. 483–490.
[18] Z. Fazlipour, E. Mashhour, and M. Joorabian, "A deep model for short-term load forecasting applying a stacked autoencoder based on LSTM supported by a multi-stage attention mechanism," *Appl. Energy*, vol. 327, Dec. 2022, Art. no. 120063.
[19] S. Lee, H. Jin, S. H. Nengroo, Y. Doh, C. Lee, T. Heo, and D. Har, "Smart metering system capable of anomaly detection by bi-directional LSTM autoencoder," 2021, *arXiv:2112.03275*.
[20] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4106–4117, Sep. 2022.
[21] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 387–395.
[22] Y. Himeur, A. Alsalemi, F. Bensaali, and A. Amira, "Detection of appliance-level abnormal energy consumption in buildings using autoencoders and micro-moments," in *Proc. 5th Int. Conf. Big Data Internet Things*. Cham, Switzerland: Springer, Jul. 2022, pp. 179–193.
[23] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," 2016, *arXiv:1607.00148*.
[24] W. Yang, X. Li, C. Chen, and J. Hong, "Characterizing residential load patterns on multi-time scales utilizing LSTM autoencoder and electricity consumption data," *Sustain. Cities Soc.*, vol. 84, Sep. 2022, Art. no. 104007.
[25] K.-J. Jeong and Y.-M. Shin, "Time-series anomaly detection with implicit neural representation," 2022, *arXiv:2201.11950*.
[26] L. Wong, D. Liu, L. Berti-Equille, S. Alnegheimish, and K. Veeramachaneni, "AER: Auto-encoder with regression for time series anomaly detection," 2022, *arXiv:2212.13558*.
[27] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Transfer learning for time series classification," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 1367–1376.
[28] T. Śmiałkowski, "INFOLIGHT SmartMeters DB, version 3," Tech. Rep., Jun. 2023. Accessed: Oct. 1, 2023. [Online]. Available: https://www.kaggle.com/datasets/tosmial/infolight-sm

TOMASZ ŚMIAŁKOWSKI received the M.Sc. degree in telecommunication engineering from the Gdańsk University of Technology, Poland, in 1984. He has many years of experience in designing digital control, telecommunications, supervision, and embedded systems. He has implemented many successful projects for electronic devices, applications, and the IoT systems. He specializes in road lighting control systems.

ANDRZEJ CZYŻEWSKI (Member, IEEE) received the M.Sc. degree in sound engineering from the Gdańsk University of Technology, Poland, in 1982, and the Ph.D. degree from the Cracow University of Science and Technology, in 1992. Since 2002, he has been a Full Professor with the Department of Multimedia Systems, Gdańsk University of Technology, where he is currently the Head. He has authorized more than 600 scientific papers in international journals, conference proceedings, and more than 30 patents. He has led more than 30 research and development projects funded by the Polish Government and participated in five European projects. He has extensive experience in soft computing algorithms and sound and image processing for applications in multimedia technology.