

Collective Contextual Anomaly Detection Framework for Smart Buildings

Daniel B. Araya^{*}, K. Grolinger^{*}, Hany F. ElYamany^{*, ‡}, Miriam A. M. Capretz^{*}, G. Bitsuamlak[†]

^{*}Electrical and Computer Engineering, Western University, London, Ontario, Canada, N6A 5B9
{dberhane, kgroling, helyama, mcapretz}@uwo.ca

[†]Civil and Environmental Engineering, Western University, London, Ontario, Canada, N6A 5B9
gbitsuam@uwo.ca

[‡]Computer Science Department, Faculty of Computers and Informatics, Suez Canal University, Ismailia, Egypt

Abstract—Buildings are responsible for a significant amount of total global energy consumption and as a result account for a substantial portion of overall carbon emissions. Moreover, buildings have a great potential for helping to meet energy efficiency targets. Hence, energy saving goals that target buildings can have a significant contribution in reducing environmental impact. Today’s smart buildings achieve energy efficiency by monitoring energy usage with the aim of detecting and diagnosing abnormal energy consumption behaviour. This research proposes a generic collective contextual anomaly detection (CCAD) framework that uses sliding window approach and integrates historic sensor data along with generated and contextual features to train an autoencoder to recognize normal consumption patterns. Subsequently, by determining a threshold that optimizes sensitivity and specificity, the framework identifies abnormal consumption behaviour. The research compares two models trained with different features using real-world data provided by Powersmiths, located in Brampton, Ontario, Canada.

I. INTRODUCTION

According to the United Nations Environment Programme [1], buildings are the largest contributor to global carbon emissions; they account for about 40% of global energy consumption and 30% of CO₂ emissions. Moreover, the World Energy Outlook [2] report states that compared to the transportation and industry sectors, buildings have the highest untapped energy efficiency potential. Hence, energy efficiency schemes aimed at the building sector can have a significant impact on achieving a green future.

One approach to building energy efficiency is to monitor energy usage with the aim of detecting and diagnosing abnormal consumption behaviour. In recent decades, modern buildings have been equipped with an increasing number of sensors and smart meters. By analyzing data from these devices, normal consumption profiles can be identified. Subsequently, when patterns that do not conform to the normal profiles are detected, the building manager is notified, and appropriate energy-saving measures are taken. More importantly, for safety-critical building services such as gas consumption, early detection and notification of anomalous behaviour (gas leakage) can help prevent potentially life-threatening disasters.

Anomaly detection refers to the process of identifying abnormal observations that do not correspond to expected patterns or events [3]. Depending on their nature, anomalies

can be broadly categorized as point, contextual or collective anomalies [3]. *Point anomalies* refer to the occurrence of a value that is considered anomalous compared to the rest of the data. For instance, an hourly heating, ventilating, and air conditioning (HVAC) consumption might be anomalous compared to previous recorded hourly values. *Contextual anomalies* take contextual or behavioural attributes into account to identify anomalies. For instance, an hourly HVAC consumption might be anomalous in winter, but not in summer. *Collective anomalies* refer to the situation in which a set of related data instances is anomalous compared to the rest of the data. Individually, these values might not be anomalous, but collectively they represent an anomalous occurrence. For instance, individually, a facility’s lighting energy consumption values might be normal as compared to previous recorded values. However, if these values are considered as a collection over a specific time window, they might represent a collective anomaly.

One of the problems of standard collective anomaly detection techniques is that there is little concern for the context of the anomaly under consideration. For example, a collection of heating energy consumption data of a school recorded every 5 minute for an hour might be anomalous in July (when schools are closed), but not in October (when schools are running).

Most modern buildings are equipped with a built-in control system referred to as a building automation system (BAS). A BAS is a part of what is referred to as an intelligent or smart building [4], and it enables building managers to automate and oversee the energy efficiency aspect of a building. By providing early detection and diagnosis of abnormal building behaviour, contextual collective anomaly detection helps not only to reduce financial cost, but also, on a larger scale, to reduce the environmental impact of electric power generation.

This research proposes a framework to identify collective contextual anomalies. The Collective Contextual Anomaly Detection (CCAD) framework uses a sliding window approach and integrates historic sensor data along with generated and contextual features to identify contextually abnormal patterns in sensor data. The framework is flexible and can adapt to requirements of the anomaly detection domain. This provides an anomaly detection platform that can be tuned to stringent or lenient requirements with regards to sensitivity, specificity

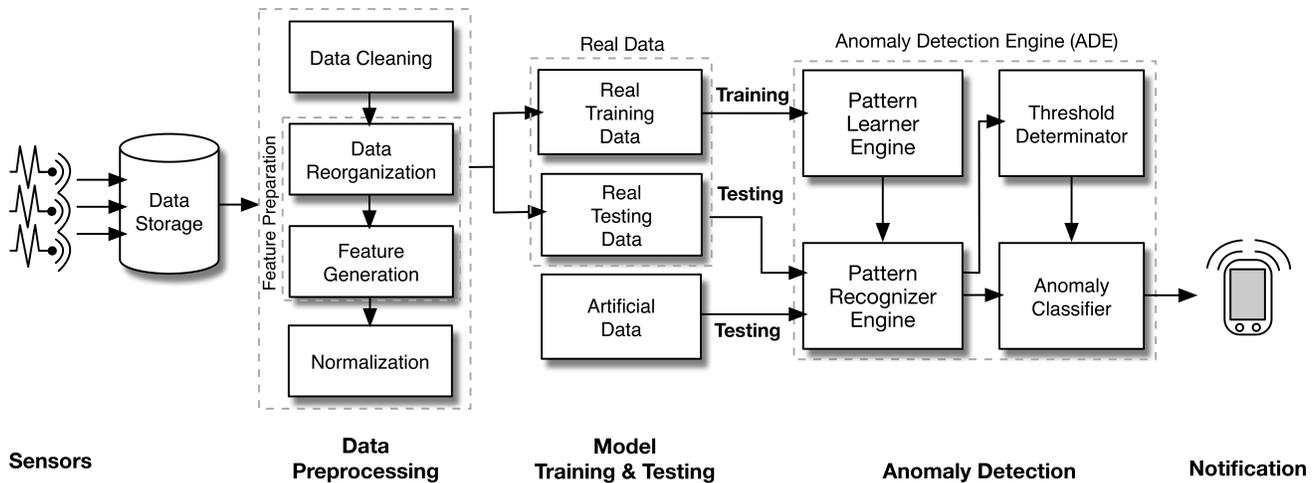


Fig. 1: Collective Contextual Anomaly Detection (CCAD) Framework.

represents a sliding hourly window. The data are organized in this way to create a pattern suitable for learning by the Pattern Learner Engine (PLE).

Feature Generation: This component introduces contextual or behavioural features into the CCAD framework. In the building energy consumption domain, the context can be spatial, temporal or weather-related. Moreover, by deriving additional sensor data features such as the mean and median, more insights can be obtained from the sliding window sensor data. The generated features are described in Table II. The temporal contextual features *day of year*, *season*, *month*, *day of week*, and *hour of day* are selected because energy consumption exhibits temporal seasonality. To ensure that the CCAD framework does not use features from other sources, weather attributes are not used in this research. The generated features (\bar{x} , s , $S_n - S_1$, $(\bar{x}_i - \bar{x}_{i-1})$, $(\bar{x}_{i+1} - \bar{x}_i)$, $Q1$, $Q2$, $Q3$, and IQR) are selected to explore whether or not these features affect the performance of the CCAD framework.

As a measure of central tendency, the mean \bar{x} of the sliding window sensor data provides a measure of the centre of the data. The standard deviation, s , gives an idea of how spread out the data are. The difference between the last and the first elements of the sliding window, $(S_n - S_1)$, shows whether or not the data on both ends of the window are the same. Moreover, the trend of the moving average between successive input data patterns is captured by including two more features: the difference between the means of the i^{th} and $(i - 1)^{st}$ input datasets, i.e., $(\bar{x}_i - \bar{x}_{i-1})$ and the difference between the means of the consumption values of the $(i + 1)^{st}$ and i^{th} input datasets, i.e., $(\bar{x}_{i+1} - \bar{x}_i)$. In a dataset that suffers from outliers, the median is a more robust measure of the centre of the data than the mean; hence, the following four features representing the median at different ranges of the sliding window are suggested: the first quartile ($Q1$), second quartile ($Q2$), third quartile ($Q3$) and interquartile range (IQR).

3) *Normalization*: To avoid suppressing the information within the smaller-valued features, the features must be normalized. The data were normalized by rescaling the features to range in $[0 \ 1]$ [16].

TABLE II: Features Generated and Domain

Feature	Description
Day of Year	1-365/366
Season	1-4
Month	1-12
Day of Week	1-7
Hour of Day	0-23
S_j , $\{j = 1, \dots, n\}$	n - size of the sliding window S - sensor consumption data
\bar{x}	Mean of sensor data values in each window
s	Standard deviation of sensor data values in each window
$S_n - S_1$	Difference between last and first elements of a sliding window
$\bar{x}_i - \bar{x}_{(i-1)}$	Difference between the means of i^{th} and $(i - 1)^{st}$ sliding windows
$\bar{x}_{(i+1)} - \bar{x}_i$	Difference between the means of $(i + 1)^{st}$ and i^{th} sliding windows
Q1	First quartile of the sensor data values in each window
Q2	Median of the sensor data values in each window
Q3	Third quartile of the sensor data values in each window
IQR	Interquartile range of the sensor data values in each window

B. Model Training and Testing

The CCAD framework uses unlabelled sensor data; hence, it relies on an unsupervised learning algorithm to identify collective contextual anomalies. The basic assumption is that historic sensor data are predominantly normal. Because of this assumption, the historical dataset (“real dataset”) can be split into training and testing data with the objective of using the testing data to evaluate the capacity of the CCAD framework to correctly identify normal behaviour.

1) *Real Training Data*: This part of the real data is used to train the Anomaly Detection Engine to recognize normal input data patterns.

2) *Real Testing Data*: Once the Anomaly Detection Engine is trained, the real testing data are used to test the specificity or true negative rate (TNR) of the model. True negative (TN) is the number of normal consumption patterns that are correctly identified; TNR is the ratio of TN and total negatives in the dataset, and it is evaluated using (1).

$$TNR = \frac{TN}{P} \quad (1)$$

where N is the number of negative instances.

To ensure a fair coverage, the test data are selected in such a way that the contextual features are evenly distributed. For instance, if temporal contextual features are considered, the test dataset is spread out to cover all months, and seasons and random hours.

3) *Artificial Data*: This research generates artificial anomalous data to test the sensitivity or true positive rate (TPR) of the model. True positive (TP) is the number of anomalous consumption patterns that are correctly identified; TPR is the ratio of TP and total positives in the dataset, and it is evaluated using (2).

$$TPR = \frac{TP}{P} \quad (2)$$

where P is the number of positive instances.

Artificial anomalous data are generated based on historic sensor data patterns. Consumption patterns can be classified into two types of period: high-activity and low-activity. A high-activity period has comparatively high energy consumption, whereas a low-activity period has either low or zero consumption values. Artificial anomalous data are generated to cover both cases. By plotting the frequency distribution of all the historic consumption data, it is possible to determine the statistically valid range of consumption values that are considered normal. This range is validated by using the 95% confidence interval. An artificial anomalous test dataset for the high-activity period can be generated by fitting an appropriate distribution to the frequency distribution plot and generating random numbers from outside the possible range of consumption values. For the low-activity period, the primary test objective is to determine whether a low-activity period’s consumption pattern behaves similarly to an active-period consumption pattern. Hence, for the low-activity period, random

consumption values can be generated from the distribution used earlier. But this time, the random values are generated from the range of possible consumption values.

C. Anomaly Detection Engine (ADE)

Anomaly detection using dimensionality reduction relies on the assumption that data contain variables that are correlated with each other and hence can be reduced to a lower-dimensional subspace where normal and abnormal data appear substantially different [17]. The Anomaly Detection Engine (ADE) proposed in this research takes advantage of dimensionality reduction and is made up of four components: the Pattern Learner Engine (PLE), the Pattern Recognizer Engine (PRE), the Threshold Determinator (TD), and the Anomaly Classifier (AC).

Algorithm 1 describes the anomaly detection engine component illustrated in Fig. 1. The `PatternLearner` is trained to recognize input data patterns consisting of historic sensor data, contextual features, and sensor data generated features (line 1).

Algorithm 1: Collective Contextual Anomaly Detection

Input : *NewSensorValue, RealTrainingData, RealTestingData, ArtificialData, ContextualFeatures, GeneratedFeatures*

Output: *Notification*

- 1 **normal_model** \leftarrow `PatternLearner` (*SlidingWindow, RealTrainingData, ContextualFeatures, GeneratedFeatures*);
- 2 **mse_negative** \leftarrow `PatternRecognizer` (*normal_model, SlidingWindow, RealTestingData, ContextualFeatures, GeneratedFeatures*);
- 3 **mse_positive** \leftarrow `PatternRecognizer` (*normal_model, SlidingWindow, ArtificialData, ContextualFeatures, GeneratedFeatures*);
- 4 **thresholdValue** \leftarrow `ThresholdDeterminator`(*mse_positive, mse_negative*)
- 5 **mse_value** \leftarrow `PatternRecognizer` (*normal_model, SlidingWindow, NewSensorValue, ContextualFeatures, GeneratedFeatures*);
- 6 **if** `IsAnomalousPattern`(*mse_value, thresholdValue*) **then**
- 7 | **return** *Notification* = true;
- 8 **else**
- 9 | **return** *Notification* = false;
- 10 **end**

Once a model has been trained with normal input patterns (line 1), the `PatternRecognizer` uses this model to reconstruct new instances of historic sensor data as well as artificially generated anomalous data. The *mse_negative* (line 2) and *mse_positive* (line 3) are the reconstruction errors of

the real testing data and the artificial anomalous data, respectively. The `ThresholdDeterminator` (line 4) function uses `mse_negative` and `mse_positive` to determine a threshold value θ , which optimizes the specificity and sensitivity of the model. Once θ has been determined, it is used to classify reconstruction error values obtained from new sensor data. The `mse_value` (line 5) denotes the reconstruction error of new sensor input data, and the `IsAnomalousPattern` function uses θ to classify a collective sensor data as contextually anomalous or not.

1) *Pattern Learner Engine (PLE)*: The pattern learner engine (PLE) uses an autoencoder [18] to train a model to reconstruct input data patterns. As mentioned in Section III A, during feature generation, to provide more information to the CCAD framework, attempts were made to introduce more contextual and other generated features. However, as the dimension increases, the engine falls into the trap of the curse of dimensionality where scalability and over-fitting issues become apparent. High-dimensional data impose performance strains on machine learning algorithms [19]. By using an autoencoder [18], which performs non-linear dimensionality reduction, the CCAD framework gains computational efficiency [20] and better classification accuracy [21] compared to other dimensionality reduction techniques such as PCA or Kernel PCA [17]. The PLE is generic and can be replaced by other dimensionality reduction techniques.

An autoencoder is an unsupervised neural network that is trained with normal input vectors $\{x(1), x(2), \dots, x(m)\}$. Assuming each data sample $x(i) \in \mathbb{R}^D$, is represented by a vector of D different variables, the input data is compressed into a lower-dimensional latent subspace to construct the outputs $\{\hat{x}(1), \hat{x}(2), \dots, \hat{x}(m)\}$ by minimizing the reconstruction error in (3) and by activating units using the activation function given in (4):

$$Err(i) = \sqrt{\sum_{j=1}^D (x_j(i) - \hat{x}_j(i))^2} \quad (3)$$

The activation of unit i in layer l is given by Eq. 2:

$$a_k^{(l)} = f \left(\sum_{j=1}^n W_{kj}^{(l-1)} a_j^{(l-1)} + b_k^{(l)} \right) \quad (4)$$

where \mathbf{W} and \mathbf{b} are the weight and bias parameters respectively; the hyperbolic tangent activation function is used in this paper.

2) *Pattern Recognizer Engine (PRE)*: Once the PLE has trained a model using the normal consumption patterns, the pattern recognizer engine (PRE) tests the model using the real testing dataset as well as the artificially generated anomalous dataset. The output of the PRE engine is a reconstruction error, which is a measure of how close the input data pattern is to the normal data pattern on which the model was initially trained. The PRE engine serves two purposes: the first is to help the

threshold determinator (TD) find a suitable threshold value, and the second is to test whether new sensor data patterns conform to the normal consumption pattern. The PRE is first used with the real testing data to determine the number of true negatives (TN). Then, the PRE is used with artificial testing data to determine the number of true positives (TP).

3) *Threshold Determinator (TD)*: This component uses the outputs of the PRE component to determine a threshold value that optimizes the sensitivity and specificity of the CCAD framework. The TP and TN values obtained from the PRE engine are used to determine the true positive rate (TPR) evaluated using (1) and true negative rate (TNR) evaluated using (2).

Lower threshold values yield higher true positive rate, but increase false positives (FP), which refer to the number of normal consumption patterns that are incorrectly identified. To find a threshold value that optimizes the trade-off between high true positive rate and low false positive rate (the proportion of normal consumption patterns incorrectly identified), the receiver operating characteristics (ROC) curve was explored. ROC is a plot in a unit square of the true positive rate (TPR) versus false positive rate (FPR) across varying threshold values. The ROC curve was chosen to analyze the performance of the anomaly detection model because, it considers all possible threshold values to evaluate both TPR and FPR of the CCAD framework. Several threshold determination techniques were explored [22] [23]. In this research, it was assumed that both specificity and sensitivity have equal weight, i.e., that the value of finding an anomalous consumption pattern and the value of identifying a normal consumption pattern is the same. However, this might not be always the case. For instance, the cost of energy wasted and/or equipment failure incurred as a result of failure to identify an anomalous incident might not be the same as the cost incurred to mobilize resources to respond to a false alarm. Using this assumption, and noting that the point (0,1) on the ROC curve (0% false alarm rate and 100% anomaly detection rate) is the ideal point, the shortest distance d from a point on the curve to point (0,1) as shown in Fig. 2 is evaluated using (5) [22].

$$d^2 = (1 - sensitivity)^2 + (1 - specificity)^2 \quad (5)$$

where d is the shortest distance from a point on the ROC curve to the point (0,1).

This distance is used to determine the threshold value that optimizes both the sensitivity and specificity of the CCAD framework. The TD component is generic, and it can be replaced by other threshold determination implementations.

4) *Anomaly Classifier and Notifier*: When new instances of data patterns are entered into the CCAD framework, their reconstruction error values are determined using the trained model in PLE. These values are then compared with the threshold θ , and patterns with a reconstruction error value greater than θ are classified as anomalous. Anomalous values trigger the notifier component to raise an alarm that notifies the

building manager, who then performs appropriate procedures.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed CCAD framework has been evaluated using a dataset provided by Powersmiths [24], a company that focusses on production of sensor devices with the aim of creating a sustainable future. Powersmiths collects various data from sensor devices, and the framework was evaluated using HVAC consumption data (kWh) of a school recorded every five minutes from 2012-2014.

Initially, incomplete and noisy data were removed from the dataset, and the dataset was reorganized so that an input instance represents an hourly sliding window data. An hourly input instance is represented by a set of five-minute interval consumption data, temporal contextual features, and additional generated features. The dataset was normalized between 0 and 1, and the final clean data size consisted of 22339 samples.

After preprocessing, the real dataset was then split into 90% real training and 10% real testing datasets and used 10-fold cross validation. Subsequently, artificial anomalous dataset were generated. To generate artificial anomalous data, historic sensor data were examined and the consumption pattern was found to have three low-activity periods; (a) weekends (b) July and August, and (c) the hours between 8 P.M. - 7 A.M. The consumption values for these periods were zero 95% of the time, so with this level of confidence, non-zero consumption values are anomalous for these low-activity periods. The frequency distribution, of the consumption values for each of the 5 minute interval consumption values of the dataset were plotted, and each of the plots have the same distribution. Fig. 3 shows the plot of a sample distribution. Because zero consumption is normal for these time periods, by excluding the zero consumption values, the distribution in the figure was fitted using the gamma distribution, and random values were generated from the range [0.1-1.2]. Again, examining the data in Fig. 3, the consumption value was found to be less than

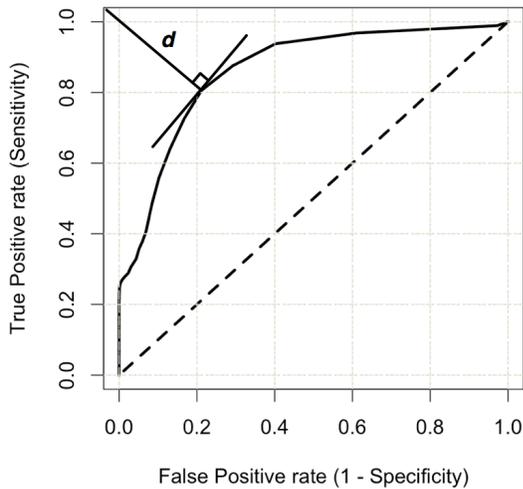


Fig. 2: ROC: optimal threshold determination [23].

1.2kWh 99% of the time. Hence, for the high-activity period, with a confidence level of 99%, random values greater than 1.2 kWh were generated using an appropriate distribution.

A. Experiments

The first step of the experiment involved tuning the parameters of the autoencoder. To learn patterns, the algorithm needs to minimize the mean squared error (MSE) of the reconstruction; hence, this metric was used to tune the parameters of the autoencoder. The autoencoder used in this research was based on an implementation in H2O [25], a scalable and fast open source machine learning platform. The experiment was performed within the R [26] programming environment using an H2O API. Both shallow neural networks (one hidden layer) and deep neural networks (more than two hidden layers) were explored. Deeper networks resulted in an increase in the processing time without a significant improvement in the MSE. Five layers were finally used; an input and output layers as well as three hidden layers. In addition, various values of regularization parameters and number of epochs were tested, and values that resulted in stable low MSE were selected. Table III shows the parameters selected.

Two experiments were then performed. The first was intended to examine the anomaly detection performance of the CCAD framework using 17 features, (5 contextual features and 12 features representing consumption data), and the second experiment was intended to explore the anomaly detection

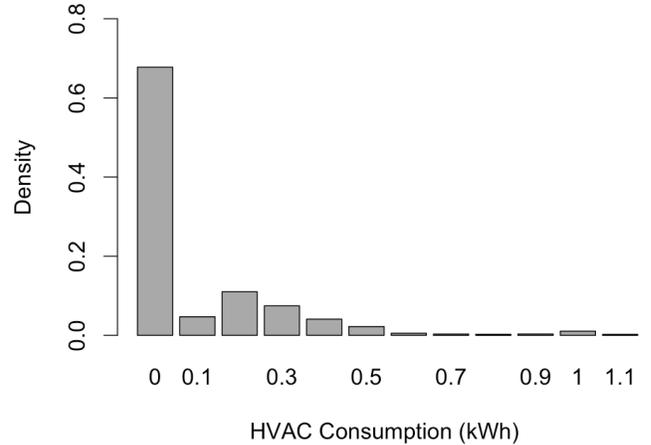


Fig. 3: Historic Consumption Frequency Distribution.

TABLE III: Autoencoder Model Parameters

Parameter	Value
Hidden Layers	3
Neurons in Hidden Layers	20, 10, 20
L1 (Regularization Parameter)	1E-04
Epoch	400
Activation	Hyperbolic Tangent

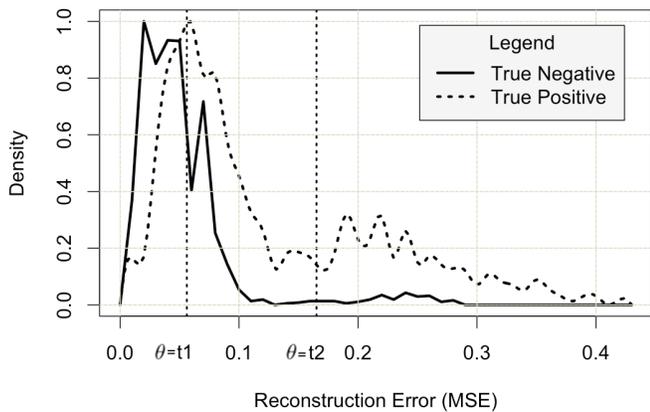


Fig. 4: TN and TP frequency distribution: 17 features.

performance of the CCAD framework using 26 features, (the 17 features mentioned earlier plus 9 more generated features, i.e., \bar{x} , s , S_n-S_1 , $(\bar{x}_i-\bar{x}_{i-1})$, $(\bar{x}_{i+1}-\bar{x}_i)$, $Q1$, $Q2$, $Q3$, and IQR described in Table II).

Experiment 1: The objective of this experiment was to examine the sensitivity and specificity of the CCAD framework using 17 features. Hence, the engine was initially trained using the real training dataset, and then first, the specificity of the CCAD framework was evaluated by testing the model using the real test dataset. Next, the sensitivity of the CCAD framework was examined by testing it using the artificially generated anomalous data. Fig. 4 shows the distribution of the TN and TP of the experiment.

Experiment 2: This was the same experiment as Experiment 1 but with 26 features. Fig. 5 shows the distribution of TN and TP for this experiment.

Fig. 4 and Fig. 5 show the trade-off between the TN (number of normal consumption patterns correctly identified) and TP (number of anomalous consumption patterns correctly identified) as the threshold θ varies. From the figures, it can be observed that as θ decreases from t_2 to t_1 , the proportion of TP to the right of θ increases, which shows that more anomalies are identified. However, as θ decreases the proportion of TN

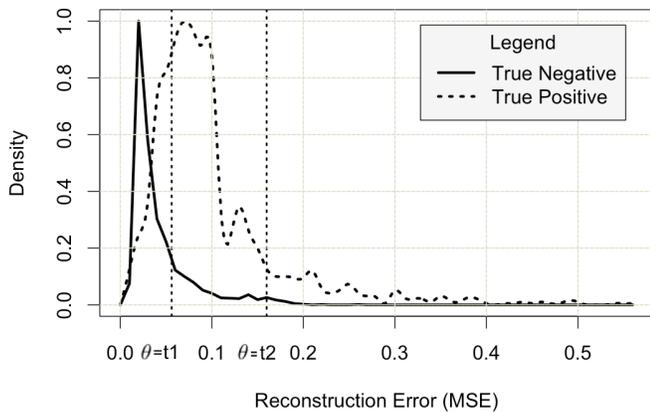


Fig. 5: TN and TP frequency distribution: 26 features.

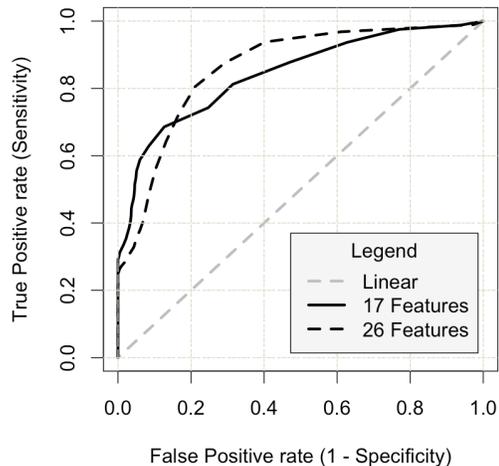


Fig. 6: ROC diagram for 17- and 26-feature models.

to the right of θ also increases, which shows that the number of normal consumption patterns misclassified as anomalous has increased. Hence, finding the value of θ that optimizes TPR (sensitivity) and FPR (1-specificity) becomes important. Hence, the ROC curve was plotted with two objectives in mind: the first was to find the threshold value that optimizes both the sensitivity and the specificity of both models; the second was to compare the performance of these two models for various thresholds.

Fig. 6 shows the ROC curves for both models. To evaluate the threshold value that optimizes the sensitivity and the specificity of the two models, the shortest distance between each of the curves and the point (0,1) was evaluated. Based on the shortest distance for the two curves, the optimal values of FPR and TPR were evaluated using (5). Table IV shows the results. By using 9 more features, the sensitivity (TPR) of the CCAD increased by 11.6%. However, its false positive rate (FPR) also increased by 8.4%.

B. Discussion

In this research, since both sensitivity and specificity are assumed to have equal weight, the results in Table IV can be seen as a slight overall increase in performance. Nevertheless, depending on the problem, either one of the TPR or FPR might be more important than the other. For instance, for critical services that have stringent FPR requirements, the 17-feature trained model performs better. Moreover, for services that have lenient FPR requirements, the 26-feature trained model performs better. Fig. 6 illustrates this clearly; for small FPR, the 17-feature curve is closer to the ideal point (0,1), whereas

TABLE IV: Model Comparison: optimal threshold values

Features	Threshold	FPR(%)	TPR(%)
17	0.07	12.7	68.6
26	0.05	21.1	80.2

TABLE V: Model Comparison: Lower Threshold Value

Features	Threshold	FPR(%)	TPR(%)
17	0.03	62.8	93.7
26	0.03	40.2	93.8

for larger and hence more flexible FPR, the 26 feature-trained model is closer to the point (0,1).

Table V shows the trade-off faced as lower threshold value is used to achieve higher TPR. By lowering the threshold value of the 26-feature model from 0.05 to 0.03, the TPR has increased to 93.8% (around 93% of the anomalous consumption patterns were correctly identified); however, the FPR has also increased to 40.2% (around 40% of the normal consumption patterns were incorrectly identified as anomalous).

Moreover, a similar lowering of the threshold value increased the TPR of the 17-feature model to 93.7%, while the FPR increased to 62.8%. This demonstrates that within the limits of the trade-off mentioned, the CCAD framework proposed can be tuned to meet the anomaly detection requirements of the problem under consideration.

V. CONCLUSIONS AND FUTURE WORK

In this research, a generic collective contextual anomaly detection (CCAD) framework has been proposed. The CCAD framework uses a sliding window approach and integrates historic sensor data and contextual as well as additional generated features to identify abnormal building consumption behaviour. The results show that this framework can successfully identify collective contextual anomalies in building HVAC consumption. Moreover, the anomaly detection rate and false alarm rates of two models (one trained with 17 features and the other with 26-features) were compared. For stringent false alarm rules and lenient anomaly detection, the 17-feature model performs better, whereas for more lenient false alarm rules and stringent anomaly detection, the 26-feature-based model performs better.

Future works will explore real-time collective contextual anomaly detection. Moreover, a comparison of the performance of other dimensionality reduction techniques such as PCA as well as the effect of different sliding window sizes on the performance of the anomaly detection metrics will be explored.

VI. ACKNOWLEDGEMENT

This research was supported in part by an NSERC CRD at Western University (CRDPJ 453294-13). Additionally, the authors would like to acknowledge the support provided by Powersmiths.

REFERENCES

[1] UNEP, "United Nations Environment Program." [Online]. Available: <http://www.unep.org/sbci/aboutsbci/background.asp>. [Accessed 19-Dec-2015].

[2] IEA, "World Energy Outlook." [Online]. Available: <http://www.worldenergyoutlook.org/weo2012/>. [Accessed 19-Dec-2015].

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[4] M. Drăgoicea, L. Bucur, and M. Pătrașcu, "A service oriented simulation architecture for intelligent building management," in *Exploring Services Science*, pp. 14–28, 2013.

[5] J. Chou and A. Telaga, "Real-time detection of anomalous power consumption," *Renewable and Sustainable Energy Reviews*, 2014.

[6] M. Wrinch, T. H. M. El-Fouly, and S. Wong, "Anomaly detection of building systems using energy demand frequency domain analysis," *2012 IEEE Power and Energy Society General Meeting*, pp. 1–6, 2012.

[7] H. Janetzko, F. Stoffel, S. Mittelstädt, and D. A. Keim, "Computers and graphics anomaly detection for visual analytics of power consumption data," *Computers and Graphics*, vol. 38, pp. 1–11, 2013.

[8] R. Fontugne, J. Ortiz, N. Tremblay, P. Borgnat, P. Flandrin, K. Fukuda, D. Culler, and H. Esaki, "Strip, bind, and search : a Method for identifying abnormal energy consumption in buildings," *12th International Conference on Information Processing in Sensor Networks*, pp. 129–140, 2013.

[9] G. Bellala, M. Marwah, and M. Arlitt, "Following the electrons: methods for power management in commercial buildings," *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '12*, pp. 994–1002, 2012.

[10] Y. Zhang, W. Chen, and J. Black, "Anomaly detection in premise energy consumption data," in *Power and Energy Society General Meeting, 2011 IEEE*, pp. 1–8, 2011.

[11] A. L. Zorita, M. A. Fernández-Temprano, L.-A. García-Escudero, and O. Duque-Perez, "A statistical modeling approach to detect anomalies in energetic efficiency of buildings," *Energy and Buildings*, vol. 110, pp. 377–386, 2016.

[12] J. Ploennigs, B. Chen, A. Schumann, and N. Brady, "Exploiting generalized additive models for diagnosing abnormal energy use in buildings," *5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pp. 17:1–17:8, 2013.

[13] A. Capozzoli, F. Lauro, and I. Khan, "Fault detection analysis using data mining techniques for a cluster of smart office buildings," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4324–4338, 2015.

[14] M. A. Hayes and M. A. Capretz, "Contextual anomaly detection framework for big sensor data," *Journal of Big Data*, vol. 2, no. 1, pp. 1–22, 2015.

[15] Y. Jiang, C. Zeng, J. Xu, and T. Li, "Real time contextual collective anomaly detection over multiple data streams," *SIGKDD Workshop on Outlier Detection and Description under Data Diversity*, pp. 23–30, 2014.

[16] D. Pyle, *Data preparation for data mining*. Morgan Kaufmann, 1999.

[17] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," *MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, p. 4, 2014.

[18] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[19] M. Bhattacharya, "Bioclimatic modelling: a machine learning perspective," in *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*, pp. 413–421, 2015.

[20] F. S. Tsai, "Dimensionality reduction techniques for blog visualization," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2766–2773, 2011.

[21] C. Orsenigo and C. Vercellis, "Linear versus nonlinear dimensionality reduction for banks credit rating prediction," *Knowledge-Based Systems*, vol. 47, pp. 14–22, 2013.

[22] K. Hajian-Tilaki, "Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation," *Caspian Journal of Internal Medicine*, vol. 4, no. 2, pp. 627–35, 2013.

[23] R. Kumar and A. Indrayan, "Receiver operating characteristic (ROC) curve for medical research," *Indian Pediatrics*, vol. 48, no. 17, pp. 277–287, 2011.

[24] Powersmiths, "Powersmiths: Power for the Future." [Online]. Available: <http://ww2.powersmiths.com/index.php?q=content/powersmiths/about-us>. [Accessed 19-Dec-2015].

[25] H2O, "H2O.ai." [Online]. Available: <http://h2oworld.h2o.ai/#about>. [Accessed 19-Dec-2015].

[26] Team, R Core, "R: a language and environment for statistical computing," R Foundation for Statistical Computing, Vienna, Austria. [Online]. Available: <http://www.R-project.org/> [Accessed 19-Dec-2015].