Unsupervised Anomaly Detection in Multivariate Time Series For Predictive Facilities Management

& Comparative Study of LSTM and Transformer Models for Time Series Forecasting.

Kartikeya Mehrotra

2211500

A thesis submitted for the degree of Master of Science in Artificial Intelligence and Its Applications

Supervisor: Dr. Faiyaz Doctor
School of Computer Science and Electronic Engineering
University of Essex

December 2023

December 11, 2023

Abstract

In the current competitive landscape, service-related sectors are increasingly harnessing the power of Artificial Intelligence (AI) and Data Science to exceed customer expectations. Cloud FM's pioneering endeavor to predict asset life and faults using electrical signal data is a testament to this trend. This paper delves into the intricacies of anomaly detection in electrical readings from UK-wide facilities provided by Cloud FM, focusing on lighting systems. This paper proposes an innovative ensemble model that integrates Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), Isolation Forest, and LSTM-based Auto-Encoder Model to robustly identify various anomaly types. This ensemble is cross-validated to enhance detection reliability and employs Gaussian smoothing to distribute anomaly probability scores across neighboring data points. Further, a comparative study between LSTM and Transformer models for time series forecasting is detailed, concluding that the LSTM-based Transformer model offers superior accuracy and efficiency. Findings suggest promising applications for such models in real-time anomaly prediction, crucial for maintenance in aviation, traffic management, and beyond.

Keywords: Facilities Management, Artificial Intelligence, Predictive Management, Anomaly Detection, Time Series, Forecasting, Neural Networks, LSTM, Transformer, Auto-Encoder, DBSCAN, Isolation Forests

Contents

1 Introduction

Today, there is no room for complacency in any sector of the world. Not in the least for the various service-related sectors, with booming technology and ever-rising competition, these sectors seem to become the most competitive of all. With there being no direct numerical perception related to services, people often estimate these performances based on how they personally feel they were tended to. This gives rise to the need for exceeding expectations. The introduction of AI and Data Science to various fields has made this need even more urgent, with clients and providers indulging in the intricacies of technology and exploring the vast ocean of opportunities within.

In one such case, is the facilities management industry, where Cloud FM wants to be the first to bring forward a huge step of innovation and science, the prediction of asset life and faults [1]. In a proactive step, they have successfully started tracking electrical signals of devices that they tend to all over the UK and are determined to pioneer this solution. This solution is novel, not just in facilities management, but across the board, from electrical to mechanical and from asset life to maybe even human life someday. This intricate use of data is the foundation of brilliant AI innovations and is the very need of the hour.

1.1 The main idea

In real world scenarios, theoretically, an asset or any equipment would show certain signs that can be associated to deterioration over time or failures. Also, in this life span, the asset will withstand certain events that may attack and shorten its life. For example, air conditioners and refrigerators are used with stabilizers to maintain a voltage that does not overload its internal components. In the absence of this, an air conditioner can receive an abnormally large amount of voltage, which could damage resistors and capacitors in any component of the machine, either resulting in an immediate failure or leaving the components a lot more fragile. Such repeat events could cause an eventual malfunction or fault in these machines and appliances. To be able to identify these events, or anomalies, can prove crucial in predicting the eventual faults that will come up. Also, by simply tracking past data, measures can be taken to predict and prevent the adverse effects of these anomalous events.

These problems may be electrical, mechanical or electro-mechanical in nature, but an indicator of all will show up in some form or another within electrical readings from various components of a device. The harmonics are one such variable in this electrical ocean that can help narrow down the sources of problems that may have occurred or may be imminent. Harmonics are voltage or current distortions at multiples of the fundamental electrical frequency. Harmonics in electrical data are critical for diagnosing power quality and predicting equipment malfunctions.

These factors are more reliable for short-term fault predictions, but may be unreliable for longer terms and life prediction patterns, where deeper concepts such as the Weibull Distribution come in. This paper focuses on these electrical readings,

captured by Cloud FM over multiple locations across UK, and further focuses on a simpler asset category, lighting systems. Compared to more complex appliances like HVACs, kitchen equipment, etc., lighting systems are a lot more binary and are easier to capture and explain. As a study of relatively novel approaches, this paper aims to understand the possibility of capturing these anomalies beforehand and predicting future time points where such anomalies may be seen.

1.2 What are anomalies?

An anomaly is classified as any occurrence that is unusual, out of the normal, and very rare. Anomalies can be of three types:

- **Contextual Anomalies:** Based on the proximity of a single point like a fan being suddenly switched on and off.
- **Global Anomalies:** A point being different from all other points within the data, such as a stone in a bag of rice.
- **Collective Anomalies:** A set of points that together perform as an anomaly, like the power consumption of a fridge left open overnight.

Even though a fan being on is not an anomaly, but a very quick on and off would lead to an unexpected surge in the power readings during an otherwise calm and null state. Global outliers, meanwhile, are based on a point being different from all other points within the data, such as a stone in a bag of rice. A collective outlier is a set of points that together perform as an anomaly, like the power consumption of a fridge left open overnight. To identify such a range of anomalies from a data set that does not contain labels for what is an anomaly, can be a humongous task. This problem calls for a model, that can work over data from different domains, helping in capturing anomalies with little to no requirement for human intervention. This paper introduces an ensemble method involving three well-known and highly regarded models to identify different types of anomalies that may exist in the data set:

- **HDBSCAN:** Identifies density-based anomalies.
- **Isolation Forest:** Targets probabilistic anomalies.
- **Auto-encoder:** Captures anomalies identified by reconstruction of data.

In real world applications, anomalies are rarely known beforehand, therefore, making the task a little more complicated, because of the lack of knowledge about what is normal and what is not. This aspect of Machine Learning is called Unsupervised Learning. The potential for such a model is uncapped, as anomalies exist everywhere, which may be useful or dangerous. Saudi Arabia defeating Argentina in the world cup is classified as an anomaly, but is a particularly historical moment for Saudi Arabia. In a similar sense, it is an anomaly for Argentina as well, but in the exact opposite sense of emotions. Data exists everywhere and anomalies occur everywhere, from purely electrical devices to purely mechanical; from solar flares to four-leafed clovers;

from weather to vehicles. A model that can potentially capture an anomaly in any system, with any data, can prove monumental, and this paper aims to progress on such a model.

1.3 Why is there a need for unsupervised anomaly detection?

Many real world applications in early stages are forced to be unsupervised due to the lack of available labelling on data to signify whether a particular point is an anomaly or not. In most cases, they must be identified through extreme research on the data and these tasks are sometimes extremely time consuming. Subtle anomalies are very difficult to be identified and are rarely even caught after time consuming efforts made on data sets.

1. For example, in a study on the hate speech detection on Arabic Social Media, [2] focused on understanding the patterns of hate speech in their data. Even though occurring more commonly than an anomaly, it was tough to be labelled automatically since hate speech can take many forms. The researchers depended on a group of annotators who worked for days, which involved discussions on what can be categorized as hate speech and what must be done if some classify it as hate and others do not, to be able to label around 5400 tweets. Regardless, tweets can be read and labelled by interpretation, but what must one do in the case of identifying anomalies in the weather patterns of a particular month based on the same month over the last 20 years? It would be almost impossible to be read by a human and labelled out and will take a team of scientists to analyse the data, calculate various statistical parameters, perform statistical tests and finally validate their findings. Such efforts will be have to made in every field where anomalies are to be identified. This is why, these anomalies are usually found or noticed after a fault occurs.

2. In 1989, United Airlines Flight 232 experienced a fatal engine failure in its tail mounted engine, due to a manufacturing defect, which caused most of the flight controls to be lost, resulting in over 100 deaths [3]. It is unrealistic to state that a model like this can capture every single anomaly in a plane, but once it was found that a mere nano-meter of a crack in the rotor blade of over 2 meters in diameters led to an inconsistency in materials over years causing it to break under the extreme stresses of an engine in flight, engineers have created tests to find these inconsistencies and calculated the life of these components a lot more carefully to avoid these events. Such a manufacturing defect can be called an anomaly and it will lead to various tests failing today.

But what can be done about anomalies that are not being checked for today? A model like the one this paper suggests should ideally capture these anomalies with the data captured in testing of these components and can potentially save lives.

1.4 The presented ideas

This paper explores the problem of capturing anomalies in unlabelled and unseen environments and comes up with a possible solution in the form of the implementation

of an ensemble model. The main aim of this paper will be to understand these hard to capture anomalous patterns and label them. To do this, the paper suggests an ensemble method involving three well known and highly regarded models to identify different types of anomalies that may exist in the data set, these being the Hierarchical Density-Based Spacial Clustering of Applications with Noise (or HDBSCAN; a progression of the well known DBSCAN model), the Isolation Forest algorithm and an LSTM based Auto-Encoder Model. This ensemble model is particularly chosen to identify various types of anomalies, density based anomalies using the DBSCAN algorithm, probabilistic anomalies using the Isolation Forest algorithm and finally anomalies identified by reconstruction of data by compression and decompression using the Auto-Encoder algorithm.

Along with this, it also explores the comparisons between two time series forecasting models, Long-Short Term Memory Neural Networks (LSTMs) and Transformer Models. The expectation of this paper, is to develop an algorithm that can successfully identify anomalous behaviours, predict future values of the time series and identify a probability of future values being anomalous. Alongside, this paper analyses the accuracy of predictions of complex time series by state-of-the-art techniques of LSTM Networks and Auto-Regressive Models to that of the relatively new concept of Transformer Models.

A model that could succeed in finding these anomalies before the fact can prove monumental, not just in facilities management, where high values of currencies are dependant, but in other cases, such as aeroplane and train maintenance and traffic management, power lines and electricity generation units, chemical researches and applications and a lot more, where there may be a dependence of life too. In such cases, all the model will need is some data, which in today's world is collected everywhere, and even if not, a small time period of data collected will be enough to identify immediate errors that may lead to faults which need to be identified urgently in the applications.

The main hypotheses this paper focuses on are:

1. An ensemble of anomaly detection techniques will successfully identify anomalies within a system in an unsupervised manner.

2. Transformer models will produce more accurate time series forecasting results compared to current market-favored LSTM models.

This paper aims to merely bring forward a potentially crucial step in anomaly detection algorithms, which will hold serious value in the future, in almost every sector of this data driven world.

The structure of this paper is as follows:

1. Introduction to the work that has already been done in the field of anomaly detection, facilities management, electrical anomalies, and the algorithms explored.

2. In-depth explanation of the model created, the process used for anomaly detection and prediction, its strengths, and weaknesses.

3. Detailed analysis of the comparative study between different time series forecasting models.

4. Interpretation of the results obtained, what can be interpreted from those results, and suggestions for future areas of focus.

2 Literature Review

This section will focus on previously done studies in the area of Anomaly Detection, Fault Prediction and similar areas that this paper is focused on. The scope for fault prediction is huge and this area is a major focus of research. This section aims to highlight the current progress made in this field and will reflect on how this paper will provide a deeper understanding into the subject.

2.1 Principal Component Analysis

Before work begins, data must be pre-processed and cleaned. Data collected from real world applications can be filled with invalid data, null values, excessive columns and missing values. In time series analysis, such as the one this paper focuses on, continuity is of major importance and data must be dealt with accordingly to avoid discontinuity. A major part of pre-processing is to feature engineering. This involves two main processes, feature selection and feature extraction. Feature selection is the process of selecting important features in the data set, while feature extraction involves compression of these features, creating new features and other possible modifications to the variables. A very exciting process here, is Principal Component Analysis (PCA).

As seen in [4], PCA is an efficient method to reduce dimensions of an otherwise expansive data set, specially in the case of anomaly detection. In their approach, they involved the use of PCA with Neural Networks to build an auto-encoder, that would reconstruct the reduced dimensional data set, which built back using the reversal of PCA would indicate the existence of anomalies. This is particularly a great use case for PCA in anomaly detection, as PCA reduces the variables but maintains a similar level of variance in the data set, producing a more concise and competent data set.

Another great application of PCA can be seen in [5], where the author indicates the importance of PCA in multivariate time series analysis, similar to the focus of this paper. This paper focuses on reducing the high dimensionality of the data set by applying PCA, and then proceed with time series analysis on the resulting reduced data set. Their paper focused on creating a clustering algorithm for the time series, and provided results in terms of both accuracy and time consumption. The excellent results in accuracy are a testament to the strengths of PCA and the fact that it reduces time consumption is a very important bonus.

PCA, in both of these cases, provides a quicker and equally strong model as compared to using the original data, and in some cases, can also improve accuracy by

eliminating redundancies in columns, and hence alleviating noise and inaccuracies, which backs its supreme importance in anomaly detection as well.

2.2 Anomaly Detection

In the world of anomaly detection, there are a few models that researchers and scientist have grown very fond of. Unsupervised detection is a very important section of these researches due to the lack of readily labelled data in real life. A few such models involve Recurrent Neural Networks, Auto Encoders, Density Based Spatial Clustering, Isolation Forest, and some other neural networks and machine learning algorithm.

2.2.1 DBSCAN and HDBSCAN

The DBSCAN algorithm is one based on the family of clustering algorithms. The basic idea of these algorithms is to cluster similar points together and to isolate those that do not fall into any cluster. The model holds a higher value above most other clustering algorithms due to its ability to capture all possible clusters without it being specified how many clusters are to be made, unlike other models like the K-nearest neighbors model, which tries to create 'K' distinct clusters based on proximity of points. Another strength of this model is its ability to handle higher dimensional data, rather than a more 2-D approach of most other clustering algorithms.

DBSCAN can be a powerful tool for unsupervised anomaly detection as seen in the case presented in [6]. In their paper, they aimed to captures anomalous flight patterns, based on airport histories. They used DBSCAN to capture outliers in the data and built an anomaly scoring algorithm on it. Using DBSCAN, they were successfully able to classify various groups of flight patterns, which may show a specific fault, failure or normalcy. The DBSCAN algorithm was successfully able to segregate normal flight patterns from different landing failures, which were due to a very high percentage of failures occurring in approach patterns of flights. With no input over the data, the model was successfully able to capture events that were flagged off as mistakes or faults, which shows its strengths in the unsupervised anomaly detection category.

While DBSCAN can identify clusters of a certain density, a more powerful tool is needed at times, when a data set may have varying densities. For this, Hierarchical DBSCAN was introduced in [8]. This model is now capable of capturing various clusters of different densities. The superiority of this method can be seen in [7], where the two algorithms, and some others based on DBSCAN and the K-means algorithm were put to test in a comparative environment to capture the abilities of these algorithms to successfully differentiate between clusters. Not only does the HDBSCAN produce significantly higher accuracy than other models, it also does this with the least time complexity compared to the other models, proving that it is a more efficient model in terms of both, predictive results and computational speed. Despite its enhanced capabilities in handling varying density clusters, HDBSCAN

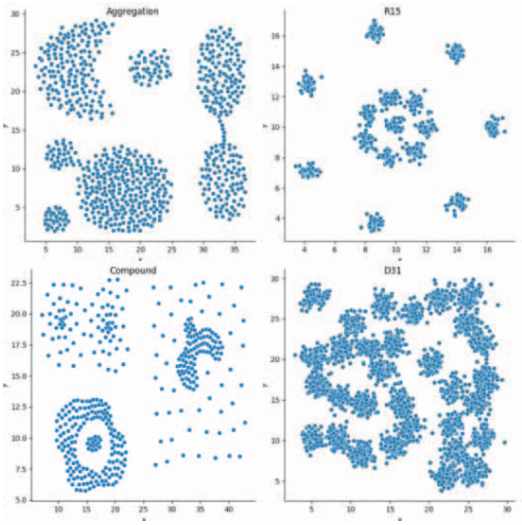

Figure 1: Clustering by DBSCAN

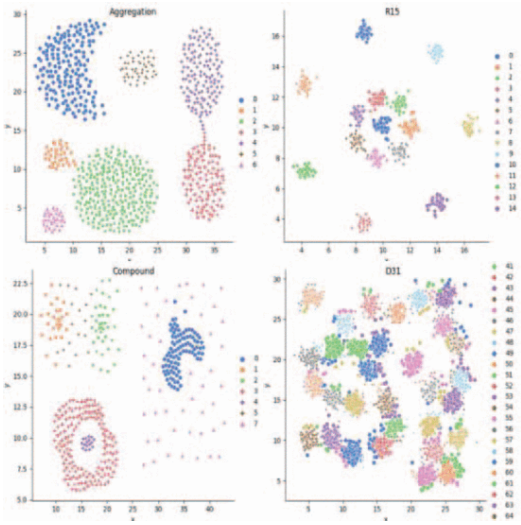

Figure 2: Clustering by HDBSCAN

Figure 3: The ability of HDBSCAN to differentiate between different clusters better than DBSCAN over 4 different data sets [7].

remains relatively underutilized in the field, possibly due to its lesser-known status and the dominance of more established models like DBSCAN.

Another application of HDBSCAN can be seen in the paper in [9], where the algorithm was used in a semi-supervised environment to capture anomalies in the power generation control systems. As explained in the paper, a power generation system is controlled by various control loops that can be vulnerable to cyber attacks. Such attacks could be detrimental as can be seen through various cases of cyber attacks on power generators.

This paper proposes a semi-supervised algorithm that is built over HDBSCAN to capture anomalies within the data from these control loops, which could then be subsequently used to prevent these cyber attacks. To achieve this, they used an HDBSCAN provided cluster as an initial point and provided it with a smaller subset of data comprising of points misclassified and those in close proximity to them. This subset was then again run through HDBSCAN and a better result was obtained. This loop was continued till the right clusters were obtained and the clusters were updated accordingly, and the final model was saved. This method showed a 100% accuracy over various labels, showing that with minimal intervention, the HDBSCAN algorithm can prove to be completely accurate in its detection and clustering.

2.2.2 Isolation Forests

A well known algorithm in the Machine Learning space is the Decision Tree. A Random Forest algorithm is one that builds up multiple such trees and provides a result based on their cumulative results, looking for majority in classification and producing averages in regression. Another well known model built on Decision Trees

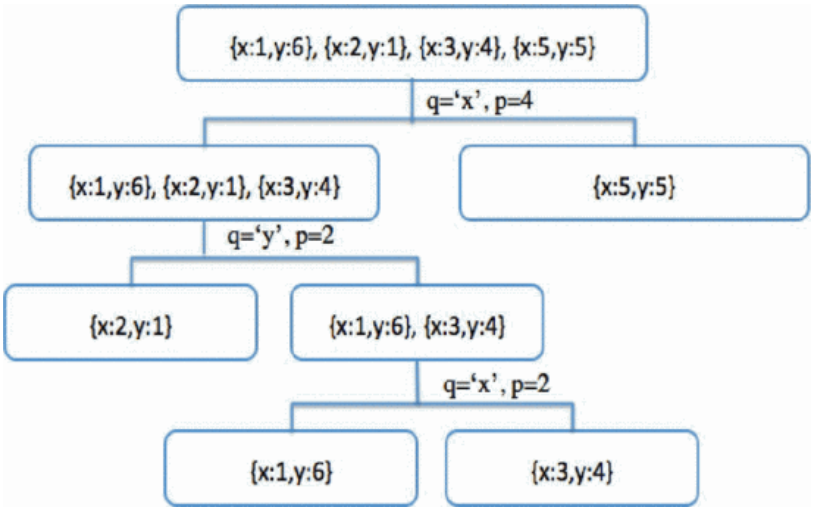

Figure 4: Working of a tree in the isolation forest. [10]

are Isolation Forests (iForest), originally introduced in [11]. Similar to the decision tree in its basic sense, the model creates splits based on values but dissimilar from the decision tree, it tries to isolate anomalies by finding the points of splits where probability is unexpectedly low but there exist data points [4]. This algorithm has been delved deep into, for researches and real world applications alike.

As seen in the case presented in [10], the isolation forest was tested against a labelled network management data set to perform anomaly detection in an unsupervised manner. With various methods of pre-processing, the iForest model proved to work extremely well, achieving true positive rates of above 97% and limiting the false positive rates to under 1.5%, in data sets where anomalies were between 0.5% to 4.3% of the data. This result shows that an unsupervised iForest can achieve extremely high values of accuracy in anomaly detection, which proves the strength of using a probabilistic model to segregate anomalies from normalcy.

Another well rounded example for iForest detecting anomalies in time series is presented in [12], where over 70 algorithms were compared over 30 data sets, and the results showed that iForest was the most consistent in the unsupervised algorithms and moreover showed better results that some supervised algorithms too. For multivariate time series analysis, it is very important for an algorithm to be time efficient, and iForests are also shown to be highly time efficient within these tests, providing no Out of Memory errors and being faster than most other methods.

The iForest method is proven to be one of the strongest unsupervised methods, but as it stands with unsupervised algorithms, it is rarely easy to verify results, which is why the data sets are unlabelled in the first case. This paper proposes that these methods be used in an ensemble algorithm to improve results by the algorithms providing cross validation to each other. But, by general consensus of the Machine Learning community, there is one algorithm that may be the best poised for detecting anomalies, that being the Auto-Encoder algorithm.

2.2.3 Auto-Encoders and LSTM based Auto-Encoders

The auto-encoder (AE) algorithm works on the most simplistic principal of compressing data and regenerating it, to see if the regenerated results are the same as the originals. Of these decompression results, the points with the highest errors are identified and labeled those as anomalies. AEs have been used far and wide for anomaly detection and have become a state-of-the-art algorithm for this use case, and to improve it, there is the LSTM based AE model. An LSTM model is a deep learning model that has the abilities to learn trends over long term and further concentrate on smaller differences with a shorter term memory, creating a strong model to analyse time series.

The findings in [13], prove the strengths of AE models, both with and without the intervention of a deeper neural network model. In this paper, a Conventional AE model was compared with a Convolutional AE model. A Convolutional AE model would be a deeper model than a conventional model since it would include a Convolutional Neural Network layer, which works on the concept of applying smaller filters over data to make the input data smaller, by applying a function over it to retain maximum information while converting all the values inside the filter to a single value. The paper here shows the conventional AE model being a very strong model, showing better results than proven clustering and anomaly detection models like k-NN, Support Vector Machines (SVM) and Triangular-Area based Nearest Neighbors (TANN). The conventional model got false positives of 4.09% and a detection accuracy of 95.85%; while the convolutional model got false positives of under 3.5% with detection accuracy of 96.87%. This showed a high level of reliability on the models, showing that the injection of a deeper neural algorithm in the AE can push it to be a lot stronger than the generic one.

In the case of this paper, an LSTM based AE was built, to further aid in time series anomaly detection as an LSTM model is best equipped to handle sequential data with its memory banks. This can be backed up by the findings in [14], where an LSTM based AE was used to capture anomalies in the carbon contents of air, captures as a time series. The main aim of their paper was to build a deep AE model that could compress and decompress carbon readings of air and provide a reconstructed value with an exclusion of anomalies. The model resulted in providing 99.5% accuracy score with a 100% precision rate and a 89.9% recall rate, proving the expertise of the model in the field of anomaly detection in time series data.

Another example is the one presented in [15], multiple models were tested against two data sets, one focusing on Industrial Gas Turbine burner-tip thermo-couples, where an anomaly can cause a sudden rise or drop in readings and the other on an Amazon EC2 CPU-consumption data set, which had CPU usage recordings, where anomalies can cause an unwanted surge, drop or shut off (outages) in the EC2 instance. In this paper, a Variational AE, a Bayes probabilistic based state-of-the-art AE model, is compared to an LSTM AE and few other benchmark models. In the results presented by this paper, over the two data sets it can be noted that the VAE is the strongest algorithm compared to every algorithm other than the LSTM AE.

This crucial comparison between the SOTA VAE model and the LSTM AE model shows the edge the LSTM model holds over the other, being equal in performance on one data set, where both models achieved an F1 score of 95%, while it far surpassed the VAE model on the other data set, where the VAE achieved and F1 score of 82% over a precision score of 86%, the LSTM AE saw an F1 score of 97% over a precision score of an outstanding 99%. These results show the efficiency of the LSTM model in handling time series data and capturing anomalies within this data using a reconstruction.

All three models discussed here, hold a distinctive edge in their respective areas of anomaly detection, namely, cluster based, probability based and Deep Neural Network and reconstruction based. These models, together can help identify anomalies generated by different analysis and representative of different patterns, resulting in a reliable ensemble method in the case where anomaly labels do not exist for verification.

2.3 Time Series Forecasting

Time series forecasting is of crucial importance to researches, scientist and other professionals today as it is a field of ML and AI that provides a deeper understanding of future events. Time series are data sets that are recorded over a time period and have been an area of key interest for ages, from the dawn of the butterfly effect for tracking weather patterns in the form of the Leibniz equation to the more recent advances in medical studies based on time dependant patterns observed in patients and subjects.

2.3.1 Long Short Term Memory Neural Networks

With the prominence of Recurrent Neural Networks (RNNs), a deep learning algorithm that worked with a focus on short term memory, there was the emergence of a more enhanced model, the LSTM, which had the ability to retain both long and short term patterns to provide a more robust understanding, specially in cases with sequential data. This architecture, in turn, has been used extensively in the field of time series analysis in general, from anomaly detection to time series predictions, both in classification and forecasting models.

The model explained in [16] explored the LSTM architecture for time series forecasting in two very innovative approaches. In the first case, they used an LSTM to forecast the energy consumption values in fridges, where multiple other variables were captures including internal and external temperatures, compressor pressure, and over hundred more variables. The aim of the first case was to compare the LSTM forecasting with that of other models that exist in the forecasting domain, like the Back Propagation Neural Network and a few Auto Regressive models. In this case, it was noticed that the LSTM had a much lower Root Mean Squared Error compared to the other models, specifically 19.5% compared to 55% and higher in the other models. These numbers show that the LSTM is much more adept in handling time series compared to other models very prevalent in the industry.

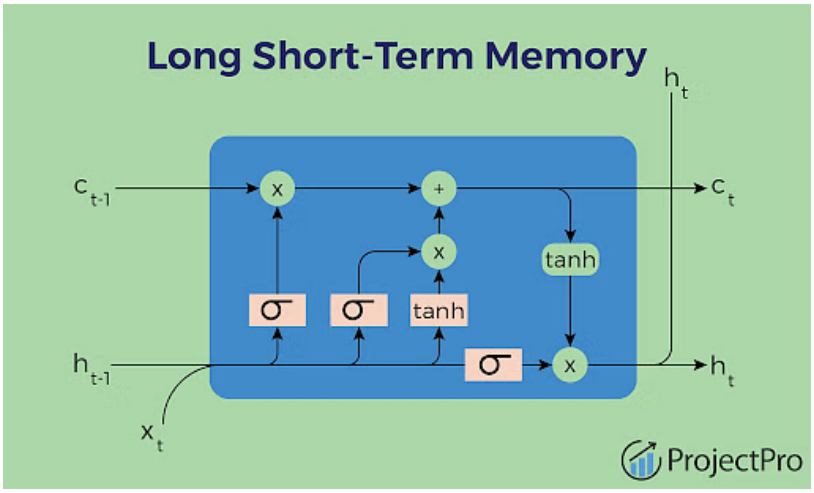

Figure 5: Working of an LSTM Model [18].

In the second case study, certain variables very graded on importance and were labelled as secondary variables, and the model was trained to forecast the future with an absence of some of these variables to mimic real world scenarios where not all cases will have all data recorded. This experiment returned satisfactory results, showing that even with the lack of some very important variables too, the model can provide decent results, and shows scope for better results with better data preparation. These results brings forward a strong edge that the LSTM model holds over other models and shows its supremacy in handling sequential data, specifically time series data, where the data is not just sequential but also very evenly spaced, such as the case of this paper as well.

In a paper focused on forecasting power loads, [17] showed the prowess of LSTMs compared to other models previously designed on the same data set. Their data set focused on load prediction at generation sites under an initiative to shift the usage of coal in indoor home heating systems onto electricity, which led to the rise of power generation loads, coming out of wind farms, in the city. The model learns from the most erratic location in the data set for adaptability over other more calm months, and by doing this, the model achieves a room mean squared error of 12.89, which is less than half than the previous experiment they conducted that involved a regression model. The accuracy score of this model was reported to by 95%, which shows a very high predictive result, which is considered acceptable for most real world applications.

These results show the practicality of LSTMs and the ability of this model to predict time series and sequences according to requirements for real world applications. Time series forecasting can be very demanding and these results cement the value of this algorithm.

2.3.2 Transformer Models

Transformer models are a novel approach to solving the problem of a lack of the model understanding context. The transformer block includes an attention module

that learns contextual values based on importance, hence building a stronger understanding of the data fed to it, and in turn making the model faster than its RNN counterparts, including LSTMs. From being first developed in 2017, it has become one of the strongest machine learning models to date and is now the focus of attention in various fields of applications, including time series.

In a paper presenting a survey of the validity of Transformer Models in time series analysis, [19] presented a study on different types of transformer models. In their paper, they aimed to analyse the impact of different transformer models on time series data coming from the benchmark data set o ETTm2 that logged electricity transformer temperatures from multiple locations in China at a 15 minute interval. The models tested involved various types of data modifications provided in pre-processing alongside the transformer models and some transformer models with architectural modification. These models were tested on two data samples, with and without seasonality decomposition, and it was noted that all models performed better with seasonal decomposition and further it was noticed that the model built with Fourier Transformation of the data and the 'vanilla' transformer, or the generic transformer model, performed the best. The model with the Fourier transform proved to be the best without seasonal decomposition for shorter forecasts, providing a root mean square error of 45% in shorter forecasts and 328% in longer forecasts. The generic model in contrast showed better results for long term forecasts with an RMSE of 267% and a little lower than its counterpart in short term forecasts with 60% RMSE. Meanwhile the models when working with seasonal decomposed data showed much better results, with the Fourier Transform based model achieving a score of 20% to 42% from short to long term forecasts and the generic model achieving scores of 20% to 53% over the same. These results go on to show that the transformer models are a lot more adept in providing shorter term results due to the attention mechanism, which can often lead to a loss of long term trends. Regardless, these results go on to prove that the Transformer models in general, when applied to the right versions of data, can go on to provide excellent future forecasts.

Transformers are a very novel approach, having been developed in 2017, and have grown in popularity due to the diverse strengths of these models. These models have been proven as adept in capturing intricacies of data and have been successful in versatile applications. This paper brings forward an application of this model, combined with an LSTM architecture for it to capture the trends of a time series, by focusing on contextual understanding with a combination of long and short term learning, making the model more robust in performance, while reducing computational loads for further real time applications.

3 Methodology

This section will bring forward the entire model architecture that has been suggested, along with a comparative study on LSTMs and Transformer models for time series forecasting, in the order of their application. The proposed architecture is an ensemble

model with an outlier detection ensemble algorithm, followed by an anomaly scoring algorithm, finally finished with a forecasting model to predict said anomalies. This architecture is aimed at providing an all round outlier prediction method for diverse multi-variate time series, to be applicable in cases with little to no knowledge of outliers.

3.1 Preliminary Understanding

3.1.1 Pre-processing

Pre-processing of data is a crucial step in any algorithm as the input scales, covariance and variable importance, all determine the model's predictive capabilities. Pre-processing involves feature selection, modification, extraction, scaling, normalisation and any other modification made to the input data.

1. Standardisation: Standardisation is the process of converting the data to a standard distribution, with the mean moved to 0. The data is scaled down and is distributed around the mean at 0. This is done by subtracting the mean from each value and dividing the result by the standard deviation as shown in 1. This results in the data being scaled down to a smaller value and the standardised data frame later helps the model to capture more nuanced trends in the data by associating each value to a Gaussian curve, hence increasing the learning of the model. Due to its nature of relying on mean and standard deviation, this procedure is affected by the presence of outliers.

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Equation 1: Standardization

2. Normalisation: Another method of scaling data is normalisation. In this method, the variables are scaled to values between 0 and 1, which is achieved by a scaling technique that reduces values using the minimum and maximum, shown in 2. This method is less sensitive to outliers as this scaling method only reduces the scale of values without modifying the distribution of the data.

$$z = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

Equation 2: Normalization

3. Feature Reduction: An important step in data processing is feature reduction. In models where numerous variables are available, the model may pick up

erroneous patterns from variables that do not hold enough importance. To handle this, features must be carefully selected and those not important enough must be dropped. This can be done using correlation analysis. Correlation analysis can be used to determine which columns are highly related to each other of which one can be removed. That is saying that two columns that increase and decrease together will indicate to similar understandings, that can be gained from either one of them. Correlation analysis hence helps remove one of these columns to reduce the impact of unwanted patterns that could be learned from these columns.

4. Feature Engineering: Feature engineering can be an impactful step in pre-processing as it is the stage where maximum information is extracted from the raw data. In this step, new features are generated, multiple features are grouped together and a data set with least redundancy and maximum information is aimed to be generated. In time series analysis, one very important factor can be time related features, such as day of the week, time of the day, week number, etc. These features help gather more understanding about the data at hand and create new correlations for the model and its predictions.

5. The PCA Algorithm: PCA is an algorithm that works on retaining a high amount of information based on variance, while compressing multiple variables into fewer variables. The PCA algorithm creates a K-Dimensional space, where K is the number of variables in the original data set, creating a random cluster of points in this K-dimensional space. This cluster is then centered onto the K-dimensional space, using a simple mean centering algorithm, where the mean becomes the new origin of the data set. This cluster of points centered on the mean is then allotted a least distance based axis, i.e., an axis that passes through the centre and has the least mean distance from each point in the cluster. This line is called the first principal component or PC-1. A second axis, PC-2, is drawn perpendicular to PC-1, and the K-dimensional cluster is now represented in a 2-dimensional space [20].

6. Applying PCA for dimensionality reduction: The process to reduce dimensions is essentially the same as a normal PCA algorithm, with an extended step before. Instead of applying the PCA to the data itself, the algorithm is applied to a covariance matrix built on the data set, which is a diagonal matrix that contains the covariance of every feature against every other feature [21]. This matrix is then passed to calculate the principal components. A threshold is set, which is the minimum required variance to be maintained, and principal components are calculated till the threshold is maintained. The calculation of covariance is dependant on a linear algebraic concept of eigenvalues and eigenvectors, which are a method of calculating linear transforms.

- An eigenvector is a specific vector transform which is created by a linear transformation, that causes no rotation, but only a change in magnitude [22].
- This change in magnitude can be achieved by multiplying a value to the original vector, or the eigenvalue.

Once the eigenvalues are calculated for each principal component, the result of

dividing each by the sum of all eigenvalues provides us with a percentage of variance that each eigenvector carries, following which, columns of lower variance are dropped based on the allowance provided by the threshold specified. Due to the components being re-scaled, the normality of the components is lost, resulting in the need for further normalisation before being processed by the models.

Anomaly detection is a crucial task and it is important that data is effectively processed for the models to capture anomalies in the best possible way. This need is further magnified by the lack of labelled anomalies, making the anomaly detection even more intricate and important to the task at hand.

3.1.2 Anomaly Detection

Anomalies in this approach were captured using clustering, probabilistic and reconstruction based models. An ensemble method is presented, that utilises all of these models to create a labelling system, which is used to create an anomaly probability score to create anomaly labels and predict them accordingly.

1. DBSCAN: The DBSCAN model, which bases its clustering on density of points. The model converts a higher dimensional data set into clusters by associating it to similar points, locating high density groups and marking them into the same cluster [23].

The model has two major hyper-parameter requirements, the epsilon and minimum points.

- Minimum points is the minimum required points within a close neighborhood for the points to be marked as a unique cluster.
- Epsilon is the minimum distance between two points for them to be labelled as neighbours.

The model then classifies each point as either a core point (a point with the minimum points in its vicinity), a border point (a point with at least one core point in its vicinity) and a noisy point (a point that has no core points within its vicinity). Over these, there is a distance metric, such as in our case, Euclidean distance, that is used to calculate the distance value between points, which must be less than or equal to the epsilon value for the point to be allocated a cluster.

The DBSCAN model cannot detect clusters of varying densities and will cause some clusters of lower densities to be labelled as anomalies. To avoid this problem, the HDBSCAN model was proposed which can identify clusters of varying densities to capture anomalies better than its predecessor.

2. HDBSCAN: The HDBSCAN model is an adaptation of the DBSCAN model, which follows a hierarchical structure to be able to adapt to varying densities. In an unlabelled and unsupervised environment, noise may arise in different sections of clusters. To be able to identify these clusters, it is important that not just one density is observed, but a varying density system must be ascertained. The HDBSCAN algorithm is able to identify such varying clusters, with a focus on not just the minimum

points used to allocate a cluster, but also a focus on the minimum points around a specific point for it to be labelled a core point. This algorithm also uses a maximum epsilon value over a fixed epsilon value, with a clear focus on creating smaller clusters and then clubbing them together as the epsilon value is allowed to grow. This allows for not just a dynamic approach to finding clusters, but also for a cluster radius to be variable (allowing for a cluster perimeter to be of the shape of an oval over specifically a circle), allowing for a stronger approach for alienating the outliers in an algorithm. The algorithm works as follows:

1. **Clusters:**
   The model starts off by creating clusters for each point in the data set, creating circles of random sizes, given that the maximum radius, called the core distance (d_c) of such a circle is not greater than the specified maximum epsilon.

$$d_c(x_p) = d(x_p, x_*) \tag{3}$$

Equation 3: Core distance between a point x_p and a neighbouring point x_*, where d is the distance function used.

2. **Hierarchy:**
   The model then implements a hierarchical structure over these clusters by varying density (epsilon value or core distance of each cluster). As the epsilon is increased till the threshold, the smaller clusters are merged within the larger ones. This hierarchy is then condensed, where more stable clusters, marked by even densities are held while those less stable are marked as noise.

3. **Cluster Extraction:**
   All the clusters are then individually labelled and the outliers are labelled otherwise.

This structure allows the algorithm to not only identify clusters more accurately, but also identifying those points that do not fall into any cluster. Bringing together the structure of a density based algorithm that focuses on a maximum distance from a point within which other points must be, and the tree like structure of a hierarchical system to club together multiple smaller clusters into one large one, discarding the need for a single central point allows this algorithm to be more specific in grouping points together.

The condensation then enables a threshold value to be used to cut off smaller clusters and points without clusters to be labelled as outliers. The entire working of the algorithm provides for a naturally more robust system in isolating outliers as well as creating very well rounded clusters.

The HDBSCAN algorithm provides a great understanding of anomalies based on density, bringing us to focus on the Isolation Forests. These models are a different type

of identifier, using a tree based system, helping capture different types of anomalies, thus helping gain a more comprehensive view of anomalies in the time series.

3. Isolation Forests: iForests are one of the most effective algorithms available for anomaly detection in high dimensional data sets. This model uses a tree structure to isolate anomalies.

Step 1: The model starts by picking random samples of the data, and randomly splits based on different values of variables to create a tree. The splitting process is repeated till a point is isolated, that is, till a point cannot be further split.

Step 2: The isolation mechanism is a simple segregation of points such that the shortest path to a point being isolated is marked as an outlier. The number of such points is limited by a hyper-parameter to the function named contamination, which takes in a percentage value and returns the smallest paths to isolation in the data set for those number of points.

Step 3: The path length to isolation is marked as an anomaly score for the point, with the highest score allotted to the smallest paths. Finally this score is averaged over multiple such trees to create a forest of isolation trees, or isolation forests, marking out the fraction of points specified in contamination, with the highest anomaly scores as anomalies.

This model due to its structure can be exceptionally quick since it does not need all samples of the data to create these trees. Along with its speed, the model's efficient structure works well with both continuous and categorical data sets allowing it to be used for any case of anomaly detection, including time series. This model is hence, one of the strongest models in machine learning to detect anomalies without the need of any prior knowledge about the distribution of data, number of classes or any specified data distribution or scale.

While the two models discussed above help capture more contextual anomalies, reconstruction based models focus better on capturing more global anomalies providing an even stronger ensemble. Auto-Encoders are one such family of deep learning models that help in capturing such global anomalies through reconstruction.

4. Auto-Encoders: AE models are strong models for anomaly detection as they work on reconstructions of the data. The models compress data and attempt to recreate the data on decompression, marking points most far apart from reconstructions as anomalies. An LSTM AE is a specialised form of AE models where LSTM layers are used for compression and decompression of data, which make it more adept to sequential learning and hence time series models.

Step 1: Data is broken into sequences of equal lengths, which serve as the sequential input for the LSTM AE.

Step 2: The input data is then encoded, or compressed into a lower dimensional representation or the latent space. This compression involves learning the temporal dependencies of the data, gaining an understanding into the data points within the data.

Step 3: The encoded layer is then decompressed in an attempt to get it back to its original form. Through multiple epochs, the model learns the data, its dependencies

and minimises the distance between the reconstruction of the data from its original state.

Step 4: After reconstructing, the output is compared to the original data sequence, recording the points most far apart between the two, marking them as anomalies. To specify how far apart a point must be, a threshold value is used, marking that fraction of points as an anomaly, such as the most separated 1% of points.

3.1.3 Time Series Forecasting

Time series forecasting is a key area of focus for this paper as the main aim here is to predict anomalies before they take place. In the realm of time series, LSTM models are considered to be the best models, but recent attention has shifted more towards the attention based Transformer models. To improve predictions this paper analyses an LSTM based transformer model. This paper also aims to compare the two models to see which is better for the current task of time series forecasting in sensor data for facilities management.

1. Long Short-Term Memory Neural Networks: The LSTM model is a deep learning model adapted from the Recurrent Neural Networks. RNNs are models developed specifically to adapt to context in a data set to better understand trends and patterns. These models contain a memory block in their architecture that enable them to retain sequential information using them. These models though, had their shortcomings, a major one including vanishing gradients. Due to retention of long term memory, often overlapping information caused a local minima of loss to be categorised as the global minima, or the vanishing of the slope beyond a local minima, hence losing out on improvements. To adapt to this problem, the LSTM model was proposed, which along with a long term memory block, includes a short term memory block. This combination hence helps the model ascertain information gains based on with long and short term contexts, making it a stronger model.

The model has a gated structure dominated by sigmoid functions which determine a value between 0-1 to be multiplied to the input, leading it to either be retained or dropped 5.

- Cell State: The memory of the LSTM. This goes through a linear function which modifies the memory only slightly in one learning cycle.
- Input Gate: The input data is passed through the input gate, which determines using sigmoid functions, the magnitude of values to be updated in the cell state.
- Forget Gate: The cell state is passed through this gate, which determines which values will be retained or forgotten from the cell state.
- Output Gate: This gate determines the output to the next state, which can be a hidden layer or the final output of the model. This contains information about the model memory, hence the information about previously passed inputs, allowing it to modify the output accordingly.

In every time step of learning, the model will update the cell state using the three gates. This allows for sequential learning and provides information retention over time, with both long and short term attention. The ability of the model to remember and forget over long time intervals promotes it to adapt adequately to the task of time series forecasting.

To further concrete a time series forecast, this paper suggests the use of a transformer model. This model handles data all at once rather than sequentially, along with an attention block. To further cement this approach, the paper suggests the use of an LSTM based Transformer model to capture sequential nuances along with an understanding of long and short term trends with a memory block

2. Transformers and the LSTM based Transformer Model: The Transformer models were built originally for a focus on sequence based data, more aligned with text modelling in Large Language Models and image based modelling due to the ability of these models to handle large data and complex patterns. These models do not focus on the order or sequence of data but rather focus more on the contextual aspects of data, making it a little less suitable for a time series model where the sequential understanding of data is crucial. To further this model into time series, a hybrid approach involving LSTMs is used. The hybrid approach will enable the model to offer higher accuracy in prediction and forecasting with a focus on both order of data or sequences and correlation of distant elements.

The working of this model involves an encoder and a decoder, but with a more focused approach on contextual learning with a parallel processing technique. With the hybrid method proposed, it is suggested to use an LSTM based encoder, apply attention blocks on the encoded input to capture deeper contextual understanding of the data and then decompress the model to generate a more complete output. The following steps will explain the proposed architecture and the workings of the model layers.

- Encoders: The LSTM based encoder will work the same as an LSTM model, using gates to compress the data to a latent space.

- Attention Blocks: The attention block creates a query, key pair, where the query is the current sequence while the keys are the other sequences to be compared to. The attention layer then generates an attention score for each query based on its dot product with its keys. These scores once normalised, are used to weigh up the original input to the attention block. This allows for each input to be modified to provide a high emphasis on the more relevant parts of the inputs. This in turn results in a contextually enhanced output from the attention block, enhancing focus on the more relevant factors of the compressed data.

- Decoders: The output from the attention block is finally decoded using LSTM layers, which further include a sequential understanding to the data and reconstruct the outputs from the attention block into the required predicted sequence length.

This architecture focuses on both the sequence and a deeper understanding of context using the hybrid architecture, providing for a more robust model to understand the patterns of the time series at hand. A combination of the above explained models should prove to be a strong ensemble algorithm to capture different types of anomalies in the time series and then further forecast them in the future providing a model that can capture and predict anomalies. The next subsection will shed further light on the proposed architecture.

3.2 Proposed Approach

This subsection focuses on the proposed architecture for the ensemble model. This approach stems from a need for accurate outlier detection in real-world data sets that were gathered from Cloud FM's extensive monitoring of electrical signals from appliances all over the UK. The following will begin with a detail on the data set, its processing and the ensemble model starting off with and ensemble outlier detection model, the anomaly score calculation and finally the proposed forecasting model. The models are very distinct yet intertwined due to their abilities to capture anomalies, which ideally must be the same presented by each model, which together provide a comprehensive solution to capturing anomalies in data sets where anomaly labelling is very tedious and difficult for manual verification.

3.2.1 Data, its Characteristics and Pre-Processing

This data set was captured by Cloud FM through an ingenious device developed by themselves, that can be installed in any electrical appliance ranging from lighting systems to fans and kitchen appliances to (Heating, Ventilation and Air Conditioning)HVAC units. The data consisted of electrical signals from various appliances from locations all over the UK. The data used was captured from various Food and Beverage outlets from Cardiff, Brighton, Haslemere, Horsham and other locations across the UK. The indoor lighting systems included ceiling lights from kitchens, washrooms, offices and seating areas, emergency lights, staircase lighting and other such setups. This provided a variety of lighting systems ensuring that patterns were captured from various differently functioning cases.

1. **Data Selection:** This data was filtered to handle only internal lighting systems, to begin with a somewhat easier task, since lights are a more binary application with less dependence on external factors such as temperatures, season, etc. This resulted in 143 unique devices, with most consisting of over 200 days of data which was evenly processed to get average values at 30 minutes apart, making each device contain over 9500 time points, with the final usable data set consisting of over 1.2 million time points.

2. **Feature Engineering:** The data consisted of over 35 variables at each time stamp, including electrical harmonics; voltage frequency, harmonics and root mean squared values; current root mean squared values; real and apparent power and more. Further, more time related features were added, like day of the

week and week number. To promote a deeper understanding of sequences, more rolling statistic values were calculated over 5, 10 and 15 time point windows, like standard deviations, means and min-max values. For another point of understanding patterns, difference columns were added, which were basically values of differences between all two consecutive time points. Further, for the more important columns selected in this case, real and apparent power, rolling absolute values were also considered.

3. **Data Detrending:** Over all these selected columns, the data was detrended. Detrending is a concept where the trend is removed from the data, as the trend can mask some important features of the data such as its seasonality and cyclic variations. Since this application was to predict a relatively short term of the future (2 days or 48 time points), it was decided to remove the trend, which would be more important in longer term applications.

4. **PCA and Dimensionality Reduction:** Before finally moving the data to the anomaly detection phase, PCA was applied to closely related columns to reduce the dimensionality of the data set. Voltage and current related columns were compressed together, harmonics were compressed together and so on. Finally, some columns were left as they were, such as day of the week, week number, power factor, true power and apparent power, since time related columns being compressed would bring little to no dimensionality reduction, true and apparent power values were very important for the anomaly detection phase and since they were unchanged, power factor would be the same regardless.

3.2.2 Ensemble Phase 1- Anomaly Detection Ensemble

The first phase of the ensemble model was focused on the implementation of an ensemble of anomaly detection models to capture anomalies and cross validate results. This phase will be crucial to the task for recognising the irregularities within the time series. The choice of the three models is backed by their diverse ability to capture anomalies. With the HDBSCAN and Isolation Forests, we can capture anomalies based on contextual understanding stemming from density and tree based isolation, while the auto-encoder captures a more global picture of anomalies in the data.

In this proposed architecture, the HDBSCAN and Isolation Forest models will focus on single 600 time point sequences, while the AE will focus on learning over 80% of the total sequences available. This is done for two reasons:

1. The HDBSCAN and iForest models will not predict as accurately when learning off an unrelated sequence to detect anomalies in a new sequence.

2. Training one model with the same hyper-parameters for all data points in a sequence will result in a model that can work with any new sequence as well.

Anomaly Ensemble- Models 1 and 2:
The first two models were built using HDBSCAN and iForest. These models were

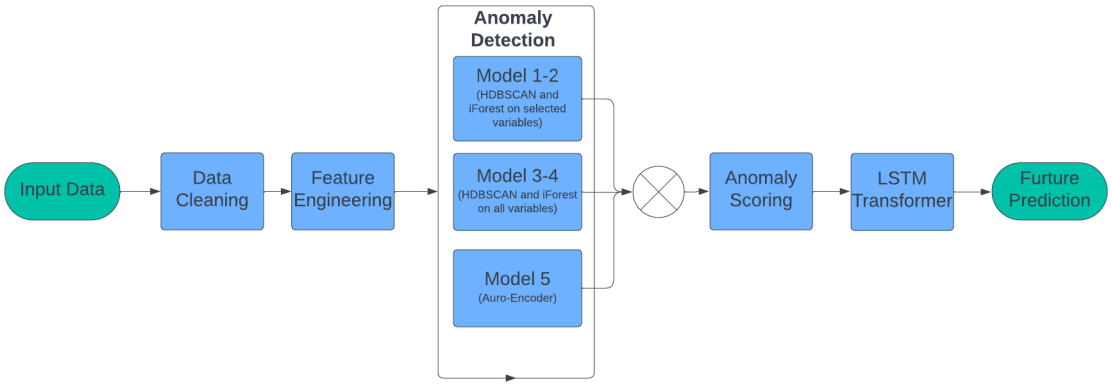

Figure 6: Proposed Architecture.

built on only time related variables (day of the week and week number) that had a high correlation to two important variables, the real and apparent power values. This was done due to the focus on the power factor being a good identifier of anomalous values. The power factor can be defined as the ratio between apparent and real power. Real power is the Wattage provided to the appliance while apparent power is the actual consumption of the appliance after various losses. This ratio is ideally expected to be greater than 0.8, stating that 80% of the power supplied to the appliance was consumed by it. An unexpected drop in this ratio can suggest there being a sudden loss of energy due to a probable fault in the device, while a prolonged loss may suggest a long term faulty device which could subsequently increase power consumption without providing adequate performance, causing a loss of money and higher wastage of power. On the other hand, an increase of this ratio above 1 will mean the device is somehow generating power in itself, which cannot be a possibility.

Anomaly Ensemble- Models 3 and 4:
The next two models were also built using HDBSCAN and iForests but were built using every variable provided after pre-processing. Both of these models are highly adept in capturing high dimensional data, which is the nature of the case analysed in this paper. Both of these models, as discussed, are possibly the strongest of their kinds and have low computational complexities. Together, these models can bring forward a significant understanding of anomalies and with future efforts can also lead to explainability of the anomaly detection process for an unseen and unlabelled database.

For all of models 1-4, the hyper-parameters were tuned using a combination of grid-searching over random samples, trial and error and instinctive parameter fine-tuning. Further, while the HDBSCAN automatically returns outliers, the iForest model requires a minimum number of outliers to be provided. For this, a dynamic method is deployed, where the dynamic contamination is taken as the ratio between the standard deviation of a required samples, compared to that of the entire data set. This value is then scaled down to a value between 0 and 0.1 to allow for a maximum of 10% of a sample to be allotted as an outlier.

Anomaly Ensemble- Model 5:
The final model in this ensemble is the auto-encoder model. This model is chosen primarily for its strengths in global anomaly detection, while the other two models were more focused on single samples in this proposed architecture, providing a more contextual anomaly detection solution. The auto-encoder, built with LSTM layers, makes it more adept in handling sequential data, and providing a more robust reconstruction, hence making it better for time series anomaly detection.

Architecture: The architecture of this model includes 7 LSTM layers, where 4 provide to encoding the data, while the 3 subsequent layers decode it. The choice for 7 layers stems out of a trade-off between reconstruction accuracy and model complexity. With different data sets, the model may need to be of varying complexity, to avoid under or over-fitting. The unit sizes and complexity was determined using multiple tests, over a cross validated train-test data set, and 10% of the training set was used for validation. The results were verified using the R2 score, which is a regression based scoring scheme. This score is calculated by subtracting the ratio of Residual Sum of Squares and Total Sum of Squares from 1, making the score of 1 perfect, while that of 0 imperfect. Residual sum of squares is the sum of the differences between squares of all predicted values and those of all true values, while the total sum of squares is the sum of squares of all the true values.

$$R^2 = 1 - \frac{RSS}{TSS} \tag{4}$$

Equation 4: R2 score, where RSS and TSS are residual and total sum of squares.

Training: The auto-encoder was trained multiple times with different batch sizes, to improve precision, using the Root Mean Square Propagation(RMSprop) optimizer and a mean absolute error loss. The RMSprop algorithm is an adaptive algorithm which aims to adjust the learning rates of each neuron in the network individually based on the averages of recent gradients for the weight.

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta) g_t^2 \tag{5}$$

Equation 5: Weighted gradient average calculations. $E[g^2]_t$ is the moving average, t is the time point and β is the decay rate.

To begin, the model was trained for 10 epochs over a batch size of 50, a very focused run, using 50 samples at a time, to increase focus by training in smaller batches from the start. The model is then subsequently trained for 10 epochs over batch sizes of 35 and 20 to further increase precision and finally a last training is done using a batch size of 15 over 30 epochs. This overall scheme allows the model to learn from the data with an increased focus on precision from the beginning.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \tag{6}$$

Equation 6: RMSprop Algorithm modifying the parameter vector. θ is the parameter vector, or the value of the neuron, while ϵ is a small constant added for stability.

With the anomalies generated from all the models, it is now important that they are scored according to a logical system of equations. To do this, the second phase of the ensemble model is prepared. In this phase, all scores are collated together to create a final score for all anomalies.

3.2.3 Ensemble Phase 2- Anomaly Scoring

In the second phase of the ensemble model, the anomalies predicted by all of the models, are passed through a linear system, to calculate a weighted score for each time point. This weighted score, is then spread out across 10 neighbouring time points, to prevent a surge of scores and to ensure that neighboring points that are predicted as anomalies also have an affect on each other.

- To do this, first, models 1 and 2 are added together with equal weightage, since they both work from anomalies predicted over power factors.
- Similarly, models 3 and 4 are also added together with equal weightage to provide higher importance to scores marked by both models together.
- These scores obtained are then weighted, arbitrarily, but with logic. The scores coming from the main 5 columns are given higher importance, while the anomalies presented from the other model are weighted lower.
- This obtained score is then added to a slightly higher weighted model 5 score, stemming from the fact that reconstruction errors are a lot more reliable compared to the other two.
- The final score would still hold the highest value when all 3 scores would agree to label a point as an anomaly, while the second highest would be when any two models agree with each other. This score gives us a very appropriate understanding of anomalies, using a strong cross validation technique to provide assurances of higher probabilities predicted.
- This final score is processed using a Gaussian smoothing kernel, which transforms a peak into a normal distribution over the specified time range, which in our case is 10. This enables, any two anomalies that lie within 10 points from each other to impact and increase each other's scores even if not directly coinciding, and allows for a range of anomaly probabilities rather than a single time point, allowing for a better predictive outcome.

This anomaly score now generated, can be used with a time series forecasting model to predict future anomaly scores, and hence capture future anomalies before they occur. To do this, the third phase of the ensemble model is undertaken, which involves the use of an LSTM based Transformer model, that is used for time series forecasting.

3.2.4 Ensemble Phase 3- LSTM and LSTM based Transformer Model

In this model, the time series sequential knowledge is prioritised. This knowledge is best accessed by an LSTM model. The model in its architecture can gain access to long and short term trends with the concept of memory blocks.

The Bi-Directional Layer- Another very important addition to the models was the bi-directional layer. The application of this layer allows the LSTM network to enhance its memory block on the basis of both past and future values rather than the conventional LSTM model that only analyses the past. The additional complexity of this model allows for an extended gain in information, providing for a somewhat contextual learning with knowledge based out of both future and present values, hence allowing for a more detailed understanding of patterns.

A more detailed understanding of patterns in the model can provide a rounded approximation of future values, helping predict the correct outcomes of anomalies in the future. Alongside, using an attention block with the Transformer architecture, can help grow an even more detailed view of patterns in the past and correct responses to current positions in the data. For example, a long term understanding can show that the data represents a sinusoidal waveform pattern, the ability to retain short term memory will further help recognise that the current point is approaching the pit of the current cycle and finally a contextual approach with the attention block will help ascertain how much lower the future values can be or if the current point is the local minima of the current cycle.

The proposed architecture, hence, allows the data to be accurately understood and predicted, which is crucial for anomaly forecasting. This architecture can further be adopted into any time series data set with very complex patterns and longer time sequences to help understand the data and its patterns with more intricacy.

Training: The model was trained for a total of 3 times, one for each of real power, apparent power and anomaly probability. The models were all trained using the Adam optimizer with a mean absolute error loss. The Adam or Adaptive Movement Estimator optimizer, is an advancement of the RMSprop optimizer, which also maintains an exponential decaying average of past gradients, or its momentum, which hence gives it the name. The model also used a dynamic learning rate scheme, where the learning rate would be slowly decreased over trained instances(an instance here is one data point being trained in one epoch). Along with this, two important approaches used were Early Stopping and Reduced Rate on Plateau. The Reduced rate on plateau function reduces the learning rate significantly if the loss for the validation data does not reduce over a few epochs, indicating in a lack of the model learning anything. The Early Stopping function then comes in, if the reductions in learning rates make no change, the model is stopped from training early, which prevents a loss

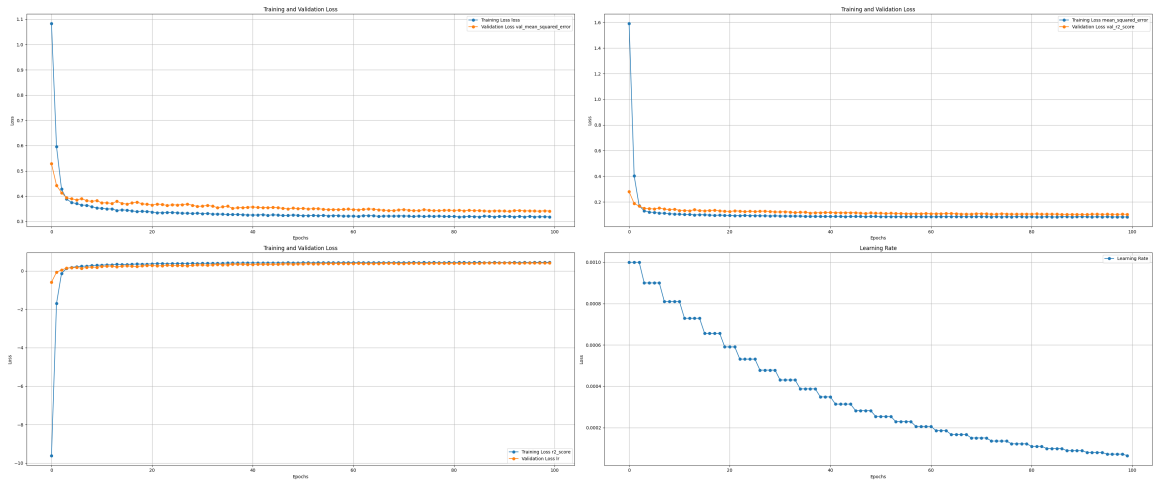

Figure 7: Scores by LSTM Model and the dynamic learning rates accordingly [Clockwise from top left, Mean Absolute Error, Mean Squared Error, Learning Rate, R2 Score.][Training in Blue and Validation in Orange]

of computational resources over a long time.

Each of these models were trained twice, to increase precision. The first training was done over a batch size of 10 for 100 epochs, providing the models sufficient learning cycles to grab very intricate patterns of the data, specially with the attention mechanism. A further second cycle was trained on a batch size of 3 for 20 further epochs. With a batch size this small, the model would not over-train and there was no evidence of under-training in the results. The model was able to produce efficient results and these values were finalised after many iterations.

4 Comparative Study

It is believed by many that the LSTM model is the best deep learning architecture that exists in the scope of time series analysis. It is a logically sound argument due to the reasons explained in detail by this paper, involving the use of memory blocks and retaining history of the model to better understand sequential models. These models hence are widely used and very highly regarded in this area. That does not mask the limitations of these models. Due to the models calculating history and maintaining memories, the learning curve of sequential data, though great for learning, is a very slow progress. These models take a long time to both, train and predict. There is also a lack of contextual learning in these models, which, with the current advancements of AI, is an integral part of most processes, be it text-based models, vision and image-based models or time series models.

These limitations call for a much needed upgrade in the architecture of deep learning algorithms to process time series. This upgrade, can come in the form of a transformer model. On analysis of time series models, this paper proves that while transformers take less time to process data, they are also faster in reducing losses

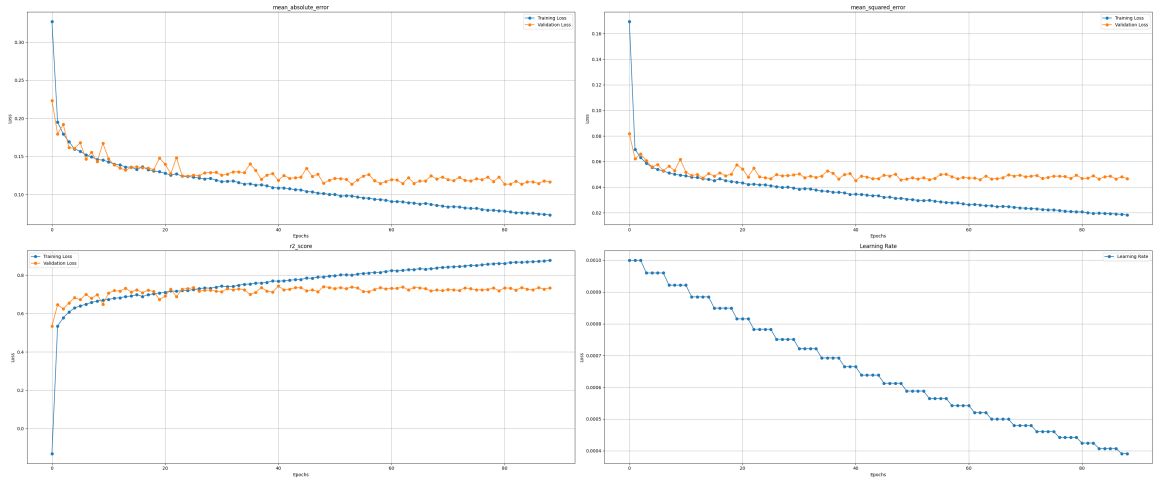

Figure 8: Scores by LSTM-based Transformer Model and the dynamic learning rates accordingly [Clockwise from top left, Mean Absolute Error, Mean Squared Error, Learning Rate, R2 Score.][Training in Blue and Validation in Orange]

Model	**Avg Time Per Epoch**	**MAE**		**MSE**		**R2 Score**	
		Train	Test	Train	Test	Train	Test
LSTM	82 secs	0.3401	0.2384	0.1032	0.0925	0.4141	0.1952
Transformer	38 secs	0.0729	0.1429	0.0183	0.0411	0.8779	0.6440

Table 1: Comparative Performance of LSTM and LSTM Transformer Models

and finding the global minima for predictions. In more static applications, both the models can be equally competitive. But in the case of applications like the one in this paper, the speed of calculations is crucial. To capture anomalies in real time, processors with very low computational power must be able to run these models, if not for training, at least for predictions. In low usage hours, like the hours of the night when a business may be closed, these models can be run over a few hours to predict the possibilities of anomalies in the next few days. Such a use case seems a lot more probable in application with a transformer model.

An LSTM based transformer model is greatly adept in capturing sequential trends and is also able to provide a strong contextual background which helps these models to predict with much higher accuracy as seen in 1. The comparative study was conducted using the same processed input data and was run for a total of 100 epochs for both models. Similar learning rate schedulers, rate reduction on plateau and early stopping checks were applied to both models. The complexity of the LSTM model was increased so as to allow it to gain as much information without over-fitting, to promote better learning at the cost of computational expense. While the LSTM model trained for 100 epochs, the Transformer model achieved a plateau at around 89 epochs and stopped early.

On discussing the comparison in more numerical terms, we can see that the Transformer model takes half the time per epoch compared to the LSTM model, showing

an extremely fast approach in comparison. The model, in addition, learns beyond the ability of a fairly complex LSTM model over 100 epochs in a matter of 2-3 epochs, and goes beyond to learn the data even better and provide much more accurate results. As seen in 7, the LSTM model starts plateauing very soon and then only reduces loss by a slight margin, staying within a very small limit of improvement. The Transformer model, meanwhile, showing significant improvement even in later stages of the learning curve [8]. The LSTM based transformer model shows a more detailed understanding of patterns and provides much better results in forecasting. The difference in prediction scores is monumental, which can be seen in the numbers presented by the mean absolute error loss, which shows a mere 0.09 mean difference in values predicted by the Transformer model, which is almost a quarter of the same metric produced by the LSTM model. Similar results are seen in both the mean squared error and R-squared metric. The R-squared metric further solidifies the findings, by showing a major increase, proving the accuracy of the transformer model to be higher than 80%.

These results show the abilities of the Transformer model in predicting time series with high accuracy. The ability of using contextual information along with the memory blocks held by the Bi-Directional LSTM layers go on to build a model that can learn more efficiently than the LSTM model, that lacks the ability to understand contexts. This attention based model has the ability to churn large data with higher efficiency, hence providing a much quicker and better result than its counterpart in LSTM. With more detailed applications, the ability for this model to forecast time series in an almost real-time fashion seem to be very realistic. With these findings, the final ensemble model will use the LSTM-based transformer model to forecast time series and capture anomalies. The final model will now aim to learn to predict all of real power, apparent power and anomaly probabilities, with the aim to build an architecture that can significantly assist the predictions of faults in the applications of Cloud FM and to help bring the community of ML and AI a step further in the bid to enable machines to process data with a higher efficiency.

5 Results

The ensemble model for anomaly detection, while being a robust model in itself, was made even more reliable by cross validation. This cross validation resulted in a high number of overlapping labels, or points where all or multiple algorithms marked a point as anomalous. This gave way to a well rounded schema for anomaly probability calculation at each point. Further, the use of a smoothing kernel for probability calculation ensured two very important factors, that an anomaly is not restricted to one time point exactly, and that anomalies in each other's neighbourhoods also increase probabilities, rather than strictly overlapping predictions. If model 1 marks a point t as anomaly, model 2 marks t-2 and model 3 marks t+2; it can be stated that the range of points has a high probability, which is being achieved through smoothing. Alongside, the deeper layer of cross-validation for contextual anomalies using 2 models each of HDBSCAN and iForests further increases confidence levels in

predictions. The following summarises the findings over 4672 samples (2.63million individual time points) in the data set-

- In model 1, the HDBSCAN model caught an average of 6 outliers per row, while the iForest model 2 caught an average of 20, with totals at 34,122 and 91,651 for each. Across the data set, a total of 15964 points were marked by both algorithms together. Further, a total of 21417 points were in proximity of each other within the two algorithms, hence providing to a higher score.

- In models 3 and 4 built out of all variables in the data, the HDBSCAN model got an average of 6 outliers, while the iForest got approximately 12 outliers on average, with totals of 27,654 and 55,501 each. 16289 points were perfectly overlapping each other, while 19891 points were in close proximity between the two algorithms

- The auto-encoder model got an average of 15 outliers per sample, with a total number of anomalies detected at 70,080.

- A total of 551 outliers are marked by all algorithms together(probability=1), while 1862 values were marked by models 1, 2 and the auto-encoder resulting in the second highest probability, equalling to 0.86. A total of 22000 points had a probability of greater than 0.6 or 60%, or approximately 0.8% of the entire data, which aligns well with the general understanding of anomalies being less than 1% of any data set. [These numbers are representative of true values as obtained from the detection models and do not incorporate the smoothing values for the purpose of capturing counts.]

As seen in the predictions in 9, 10 and 11, it is clear to see the high spikes in anomaly probabilities pertain to some very visual abnormalities within the real and apparent power values, hence further solidifying the anomaly detection ensemble. Most of the high spikes, which indicate multiple models unanimously captured the point as an anomaly, can be correlated to a sudden spike in the apparent power, or an abrupt change in the power factor. Also, as in the case of 10, the model is successfully capturing the abnormal behaviour where normally the cyclic behavior indicates the value to go down to 0 in one phase, the one time that it maintains a value little higher than 0 in both true and apparent power, the section is labelled as an anomaly.

These findings generate a high level of satisfaction in anomaly detection and help move to the next point of time series forecasting. Unlike anomaly detection, this task is not unsupervised and can be cross-validated, as is done by various metrics for performance. The LSTM based transformer model provided a very high level of accuracy scores in the form of the mean absolute error, mean squared error and R-squared metrics. These results [2] go to show the high degree of reliability on the model and its applicability in real life.

Further, even on visual analysis [9], the model seems to predict both power values adequately, but more importantly, captures the anomaly patterns exceptionally well. This, being the more crucial section of this experiment, makes the model performance more agreeable.

Model	MAE		MSE		R2 Score	
	Train	Test	Train	Test	Train	Test
Real Power	0.0711	0.1398	0.0164	0.0391	0.8843	0.6512
Apparent Power	0.0885	0.1425	0.0257	0.0525	0.8129	0.5715
Anomaly Prob.	0.0073	0.0053	0.0001	0.00005	0.9953	0.9891

Table 2: Performance Metrics of the LSTM Transformer Model

6 Discussions and Conclusion

This paper goes on to show the proposed architecture is highly promising for anomaly detection and time series forecasting. Cloud FM could use the proposed architecture for monitoring their appliances and explore the future possibility of deploying similar architectures to their use cases. These architectures, though not valid for real time, can be used to monitor devices for a longer time frame and provide predictions for a longer time period. Also, with the use of more data available to the organisation, the model trained over a larger data set, hence providing it the opportunity to learn more complex patterns and detect anomalies with more context. The outcomes of this model are promising and can be used by the organisation to build on, analysing more time points across more locations and appliances, providing them with predictions that will be more accurate and possible, more in advance than the 2-day window currently used.

With a wide scope of applications, it is an area that will surely gain a lot of focus in the coming time. This architecture though, required the focus of some more future attention. This model can be deployed in a case where there is superficial human involvement to validate anomaly labels predicted, after the event has occurred. This can prove very useful, providing a semi-supervised model for anomaly detection, with a framework based on reinforcement learning. That architecture, can both, help validate the model and help it improve over time. This can then be provided a parallel mechanism, which works to identify the patterns this model observes as an anomaly, hence creating an model that provides an understanding of anomalies, which can then auto-label points as they occurs in the future. The successful completion of this model can then provide a self-learning model, as the model labels points based on the ensemble mentioned here with involvement of the parallel mechanism mentioned, these values once predicted can then be cross verified by the parallel mechanism and the model can retrain itself to improve its capabilities of anomaly detection.

Alongside a self-learning algorithm, the proposed architecture would be best proven once a benchmark test is successfully run over this architecture for anomaly detection. Further, the Transformer model must be tested over various other applications as well, and must be compared to other leading models in those fields of time series analysis, such as the Auto Regression and Moving Averages models in financial data and stock forecasting. Also, there is undoubtedly the need to improve model efficiency for real time applications, where smaller microprocessors and micro-controllers would be the only available hardware options for applications.

Real world data, can be extremely raw and contain various possible scenarios. There is no guarantee of the data being recorded in a normally functioning environment. Various elements, coming from various locations, could be at many different stages of their life, have different running cycles and moreover have higher variability in functioning than data recorded in a stable and controlled environment This provides lesser stability and hence makes the task of the model's training harder. This proposed architecture, trained and tested over data collected in the real world, with no previous history of positively performing models, has provided results of the highest order. This ensemble architecture can prove to be the stepping stone for many future developments in model architectures to come in the spaces for both anomaly detection and time series forecasting and might also be a possible entry into real world applications with some computational bandwidth. The models explored are highly efficient and could also be a building block for future models that focus on real-time prediction.

7 Appendix

Some random predictions made by the transformer model. The top graph indicates the anomaly probability, followed by real power and finally apparent power.

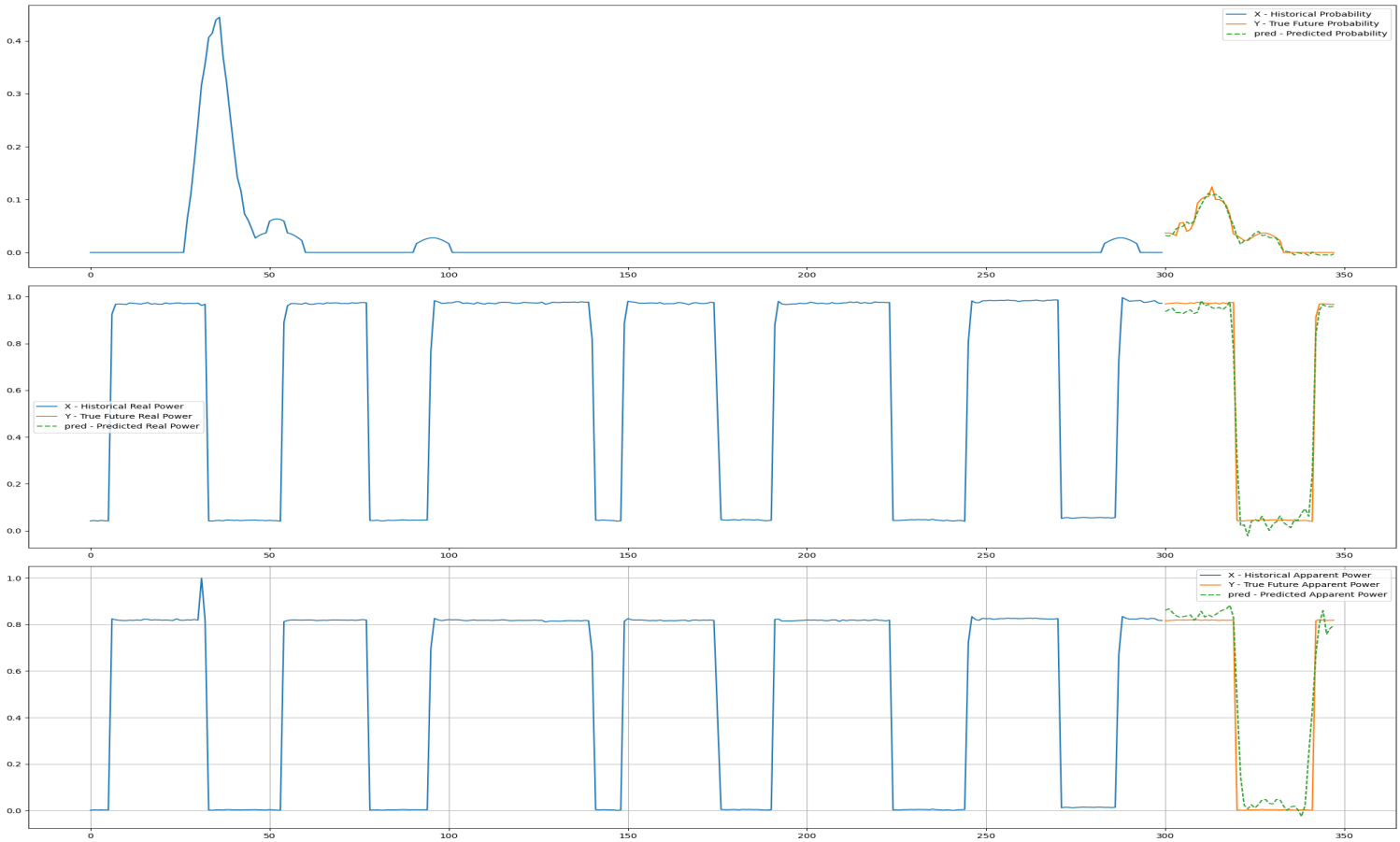

Figure 9: Random Prediction 1

Code repository: `https://cseegit.essex.ac.uk/22-23-ce901-ce902-sl/22-23_CE901-CE902-SL_mehrotra_kartikeya`

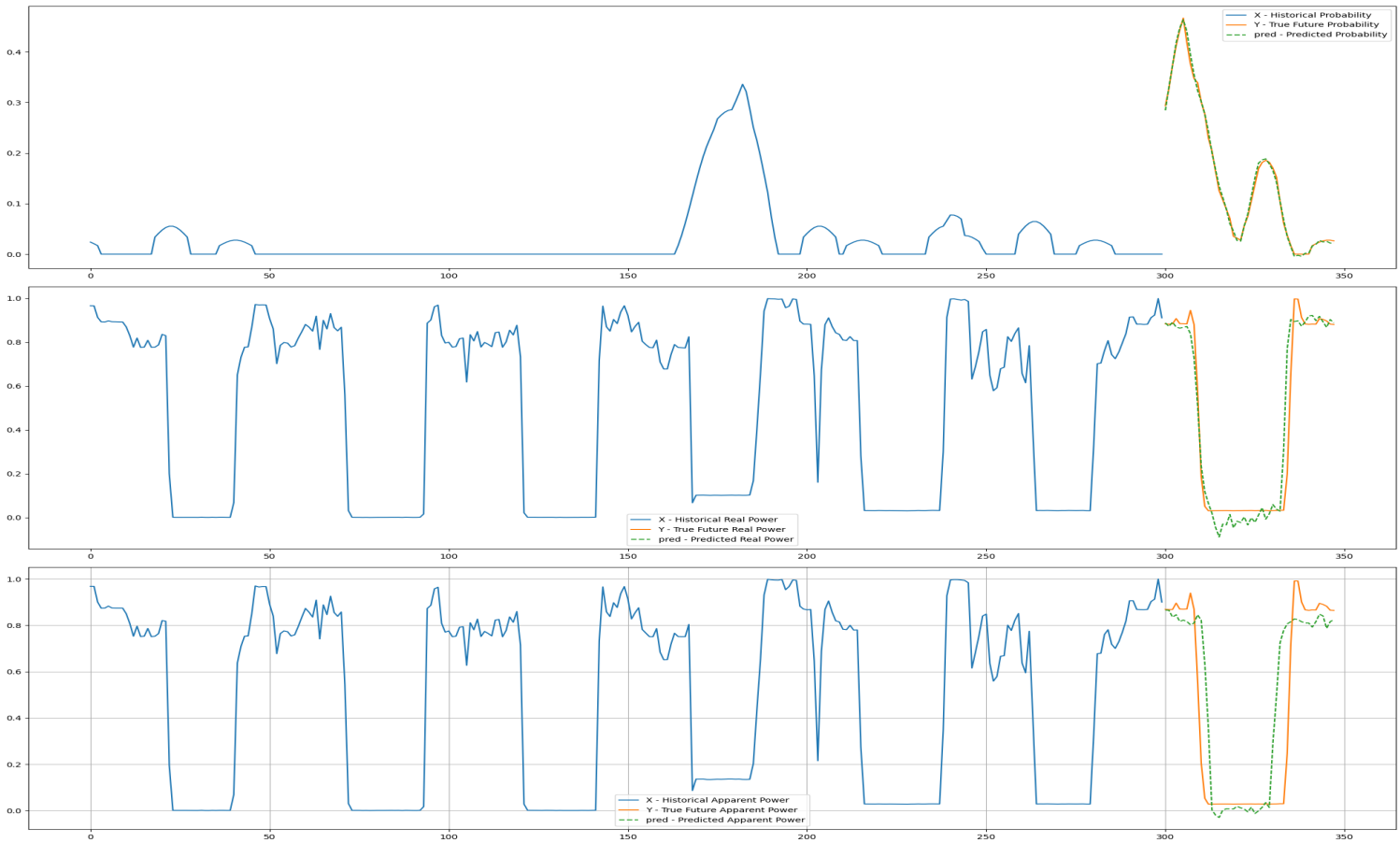

Figure 10: Random Prediction 2

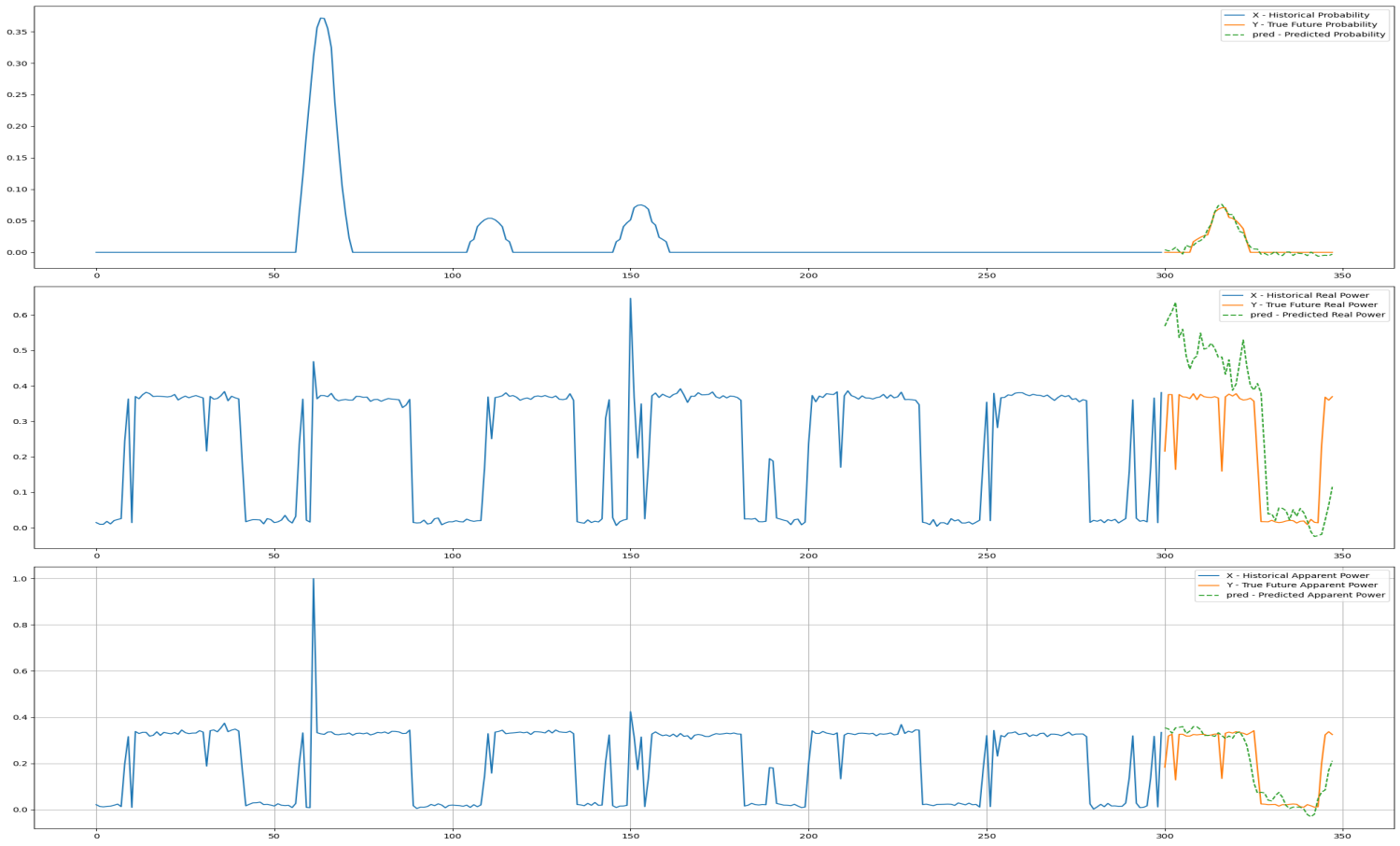

Figure 11: Random Prediction 3

References

[1] *Cloud fm*, Cloudfm House, Charter Court, Severalls Industrial Park, Newcomen Way, Colchester CO4 9YA, Jul. 2023. [Online]. Available: https://www.cloudfmgroup.com/.

[2] S. Alsafari, S. Sadaoui, and M. Mouhoub, "Hate and offensive speech detection on arabic social media," *Online Social Networks and Media*, vol. 19, p. 100 096, 2020, ISSN: 2468-6964. DOI: https://doi.org/10.1016/j.osnem.2020.100096. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468696420300379.

[3] Wikipedia, Nov. 2023. [Online]. Available: https://en.wikipedia.org/wiki/United_Airlines_Flight_232.

[4] S. Crépey, N. Lehdili, N. Madhar, and M. Thomas, "Anomaly detection in financial time series by principal component analysis and neural networks," *Algorithms*, vol. 15, no. 10, 2022, ISSN: 1999-4893. DOI: 10.3390/a15100385. [Online]. Available: https://www.mdpi.com/1999-4893/15/10/385.

[5] H. Li, "Multivariate time series clustering based on common principal component analysis," *Neurocomputing*, vol. 349, pp. 239–247, 2019, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.03.060. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092523121930400X.

[6] K. Sheridan, T. Puranik, E. Mangortey, O. Pinon Fischer, M. Kirby, and D. Mavris, "An application of dbscan clustering for flight anomaly detection during the approach phase," Jan. 2020. DOI: 10.2514/6.2020-1851.

[7] Z. He, Z. Yin, A. Chai, and Z. Bi, "Hierarchical clustering algorithm for anomaly detection on intelligent production line," in *2022 11th International Conference of Information and Communication Technology (ICTech))*, 2022, pp. 398–405. DOI: 10.1109/ICTech55460.2022.00086.

[8] R. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," vol. 7819, Apr. 2013, pp. 160–172, ISBN: 978-3-642-37455-5. DOI: 10.1007/978-3-642-37456-2_14.

[9] P. Wang and M. Govindarasu, "Anomaly detection for power system generation control based on hierarchical dbscan," in *2018 North American Power Symposium (NAPS)*, 2018, pp. 1–5. DOI: 10.1109/NAPS.2018.8600616.

[10] X. Chun-Hui, S. Chen, B. Cong-Xiao, and L. Xing, "Anomaly detection in network management system based on isolation forest," in *2018 4th Annual International Conference on Network and Information Systems for Computers (ICNISC)*, 2018, pp. 56–60. DOI: 10.1109/ICNISC.2018.00019.

[11] F. T. Liu, K. Ting, and Z.-H. Zhou, "Isolation forest," Jan. 2009, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[12] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proc. VLDB Endow.*, vol. 15, no. 9, pp. 1779–1797, May 2022, ISSN: 2150-8097. DOI: 10.14778/3538598.3538602. [Online]. Available: https://doi.org/10.14778/3538598.3538602.

[13] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *2018 Wireless Telecommunications Symposium (WTS)*, 2018, pp. 1–5. DOI: 10.1109/WTS.2018.8363930.

[14] Y. Wei, J. Jang-Jaccard, W. Xu, F. Sabrina, S. Camtepe, and M. Boulic, *Lstm-autoencoder based anomaly detection for indoor air quality time series data*, 2022. arXiv: 2204.06701 [cs.LG].

[15] S. Maleki, S. Maleki, and N. R. Jennings, "Unsupervised anomaly detection with lstm autoencoders using statistical data-filtering," *Applied Soft Computing*, vol. 108, p. 107 443, 2021, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2021.107443. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621003665.

[16] J. Q. Wang, Y. Du, and J. Wang, "Lstm based long-term energy consumption prediction with periodicity," *Energy*, vol. 197, p. 117 197, 2020, ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2020.117197. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544220303042.

[17] Z. Chen, D. Zhang, H. Jiang, *et al.*, "Load forecasting based on lstm neural network and applicable to loads of "replacement of coal with electricity"," *Journal of Electrical Engineering And Technology*, vol. 16, no. 5, pp. 2333–2342, 2021. DOI: 10.1007/s42835-021-00768-8.

[18] S. Aggarwal, *The ultimate guide to building your own lstm models*, Nov. 2023. [Online]. Available: https://www.projectpro.io/article/lstm-model/832.

[19] Q. Wen, T. Zhou, C. Zhang, *et al.*, "Transformers in time series: A survey," Feb. 2022.

[20] Sartorius-Blog, *What is principal component analysis (pca) and how it is used?* Aug. 2020. [Online]. Available: https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186#:~:text=Principal%20component%20analysis%2C%20or%20PCA,more%20easily%20visualized%20and%20analyzed..

[21] Z. Jaadi, *A step-by-step explanation of principal component analysis (pca)*, Mar. 2023. [Online]. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[22] C. V. Nicholson, *A beginner's guide to eigenvectors, eigenvalues, pca, covariance and entropy*. [Online]. Available: https://wiki.pathmind.com/eigenvector#:~:text=To%20sum%20up%2C%20the%20covariance,is%20expressed%20by%20the%20variance..

[23] N. S. Chauhan, *Dbscan clustering algorithm in machine learning*, Apr. 2022. [Online]. Available: https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html.

[24] Scikit-Learn, *Scikit learn docs*. [Online]. Available: https://scikit-learn.org/stable/modules/.

[25] Tensorflow-Google, *Tensorflow documentation*. [Online]. Available: https://www.tensorflow.org/guide/keras.

[26] Statsmodels, *Statsmodels api*. [Online]. Available: https://www.statsmodels.org/stable/api.html.