energies

MDPI

Article

False Data Injection Attack Detection in Smart Grid Using Energy Consumption Forecasting

Abrar Mahi-al-rashid [1], Fahmid Hossain [1], Adnan Anwar [2,*] and Sami Azam [3]

[1] Department of Mechanical and Production Engineering, Islamic University of Technology, K B Bazar Rd., Gazipur 1704, Bangladesh; abrarmahi@iut-dhaka.edu (A.M.-a.-r.); fahmidhossain@iut-dhaka.edu (F.H.)
[2] School of IT, Deakin University, 75 Pigdons Rd., Waurn Ponds, Geelong 3216, Australia
[3] College of Engineering, IT and Environment, Charles Darwin University, Casuarina 0810, Australia; sami.azam@cdu.edu.au
* Correspondence: adnan.anwar@deakin.edu.au; Tel.: +61-3-5227-3679

Abstract: Supervisory Control and Data Acquisition (SCADA) systems are essential for reliable communication and control of smart grids. However, in the cyber-physical realm, it becomes highly vulnerable to cyber-attacks like False Data Injection (FDI) into the measurement signal which can circumvent the conventional detection methods and interfere with the normal operation of grids, which in turn could potentially lead to huge financial losses and can have a large impact on public safety. It is imperative to have an accurate state estimation of power consumption for further operational decision-making.This work presents novel forecasting-aided anomaly detection using an CNN-LSTM based auto-encoder sequence to sequence architecture to combat against false data injection attacks. We further present an adaptive optimal threshold based on the consumption patterns to identify abnormal behaviour. Evaluation is performed on real-time energy demand consumption data collected from the Australian Energy Market Operator. An extensive experiment shows that the proposed model outperforms other benchmark algorithms in not only improving the data injection attack (95.43%) but also significantly reducing the false positive rate.

Keywords: smart grid; deep learning; auto-encoder; false data injection; cyber security; anomaly detection

check for updates

Citation: Mahi-al-rashid, A.; Hossain, F.; Anwar, A.; Azam, S. False Data Injection Attack Detection in Smart Grid Using Energy Consumption Forecasting. *Energies* **2022**, *15*, 4877. https://doi.org/10.3390/en15134877

Academic Editors: Basile L. Agba, Marthe Kassouf and Mourad Debbabi

Received: 9 May 2022
Accepted: 28 June 2022
Published: 2 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

While electricity consumption growth has slowed in recent years, it is predicted to increase significantly in the future. Peak electricity demand (times when electricity demand is at its highest) is growing at a much faster rate than overall consumption and is increasingly a challenge for the grid. Smart Grid consists of a network of measurement units or sensors which are coordinated through remote control and automation to balance the supply and demand by ensuring a reliable operation of electricity grid and benefits the environment, the community, and the consumer. It is incorporated with information and communication technologies which makes the interactions between devices more efficient [1]. It is a complex cyber-physical system which has huge advantages like identifying and isolating faulted area(s) and restoring unaffected area(s) [2]. Utilization of computing and communications effectively improves the quality of monitoring and control in the smart grid. However, it also has vulnerabilities to cyber-attacks [3].

Quick development of smart grids coupled with IoT devices has brought the issue of cyber-attacks; thus, energy companies and electricity systems face a substantial and growing threat of cyber attacks. Because the power grid is a critical infrastructure, it is a tempting target for sophisticated and well-equipped attackers. Cyber attacks are usually based on Malicious Software (malware) that must communicate with a controlling entity over the network to coordinate and propagate. From small viruses built by isolated groups to massive and complex cyber-attacks done by states and bigger entities, electrical power

stations and nuclear facilities have shown breaches that governments must deal with. For example, Ukraine Blackout compromised three electric power distribution companies affecting approximately 225,000 people for several hours. The attack was done using spearphishing and a Trojan horse called BlackEnergy which was able to delete the data, destroy the hard drives, and take control of infected computers. Not only attacking utilities equipment, it went further and launched a coordinated denial-of-service attack on the support phone number of companies who manage the power station. Consequently, users could not reach the support centers to inquire about the breakdown. The severity of a coordinated cyber attack is shown in [4]. It was the first publicly acknowledged incident, and it exposed the weakness of highly automated cyber-physical systems like smart grids. There was a significant increase of cyber-attacks in 2017 which grew even more in 2018 as described by the United States Department of Homeland Security—from Russian infiltration into the American electrical network to more than 4300 cyber-attacks of the French network (Electricity Transmission Network known as RTE). In September 2019, Dtack malware was found in India's Kudankulam Nuclear Power Plant, which was used to collect and harvest data from the victim computer [5]. One of the captured Dtrack samples contained information about the power plant's internal network. Since these cyber-attacks may lead to catastrophic events by misleading the control system, it is crucial to detect and mitigate these attacks.

According to the 2019 report of Kaspersky ICS CERT [6], injection is one of the most common vulnerabilities in an Industrial Control system (ICS). The recent literature shows that False Data Injection (FDI), a data integrity attack similar to the man-in-the-middle or spoofing attack, can severely damage the reliability of the system [3]. As illustrated in [7], malicious attacks can affect the operational decision in Supervisory Control and Data Acquisition (SCADA) of the grid by hampering correspondences between devices. A spiteful attack can create disturbances in different segments of a smart grid by blocking communication or by adding falsified information [8]. A well organised simultaneous cyber-attack on multiple substations can lead to whole system breakdown [9]. False data on power demand may lead unpleasant circumstances such as additional energy generation or lack of power, which may become a public safety issue [10]. Anomaly detection is one of the most effective ways of countering these FDI attacks [11].

The SCADA continuously monitors and controls the power flow to ensure the operation of uninterrupted operation of the grid. It contains a large amount of connectivity among the field devices which provides information for decision-making in the center for remote maintenance of the system [12]. This communication infrastructure is critical to the SCADA, but it also increases the vulnerability of the system to cyber attacks. If a FDI attack is successful in compromising data from the field devices, the SCADA will seriously be crippled. A fundamental requirement of a smart grid is to have accurate state estimation to operate properly that completely relies on the SCADA measurements. The data integrity between the field devices and the center has to be maintained all the time. However, the communication between the field devices and the center is one of the most vulnerable places for FDI attacks [13]. To neutralize cyber-attacks, a combination of Firewalls and Intrusion Detection Systems (IDS) is recommended by the Department of Homeland security [14]. The most effective measurement against FDI is to detect, prevent, and detach irregular activity of a particular system by continuous monitoring [15]. In order to overcome this challenge, two kinds of detection systems were introduced which were model-based detection algorithms and data-driven detection algorithms [16]. In model-based detection algorithms, the whole grid system is modeled with real-time measurements and system parameters, whereas the data-driven detection algorithms do not require any system parameters or models. The main disadvantage of model-based detection algorithms is that it requires system parameters, and, if there are uncertainties in these parameters, then the performance of the algorithms will be affected. Model-based algorithms are also eminently computationally complex [16].

In this paper, a data-driven two step anomaly detection engine based on a deep learning model CNN-LSTM Auto-encoder is proposed. The anomaly detection technique requires accurate forecasting on the real-time data to efficiently detect the presence of anomalous behaviours [17]. Different machine learning models and a statistical models are compared in this work to benchmark the proposed forecasting model. Experimental results show that, out of these considered machine learning algorithms, the CNN-LSTM Auto-encoder had the most accurate forecasting on the dataset, which motivated us to utilize this method for developing the anomaly detection engine. The CNN-LSTM Auto-encoder recognizes the behavior patterns of the real-time data using the historical data. Then, an optimum threshold boundary σ was empirically selected depending on the false positive rate, false negative rate, and accuracy of an anomaly detection engine considering a wide range of attack scenarios. The key contributions made in this paper are as per the following:

- proposes a real-time forecasting aided anomaly detection technique to identify a specific type of cyber attack in the SCADA system known as false data injection attack.
- present CNN-LSTM auto-encoder architecture and determine the existence of the intrusions if the field measurements deviate significantly from the forecasts.
- experiment is conducted on a real-time dataset (Australian Energy Market Operator) that shows a significant gain in performance in comparison to benchmark methods.

The rest of the paper is organized as follows: in Section 2, literature related to forecasting based anomaly detection engine is discussed. In Section 3, the proposed two step approach is described. The performances of different forecasting models and a comparison among them are discussed in Section 4 followed by the performance of the anomaly detection. In Section 5, the concluding remarks are presented.

2. Related Work

For decades, forecasting has been used as a powerful tool in weather pattern prediction, economic events, affair outcome prediction, supply chain, and more. Forecasting has been proposed to exercise in verity sectors for its efficiency, low risk management, cost reduction, and improved optimization. In this section, existing work is classified in two categories: (i) literature based on forecasting models and (ii) literature based on forecasting based anomaly detection.

2.1. Forecasting Models

Economic forecasting is becoming more widely used due to its better decision-making capabilities. Subsequently, forecasting has been used in supply chain management [18]. The goal of supply forecasting assists customers by supplying the right products at the right place, time and price. Another approach of forecasting is prediction of financial time series using support vector machine (SVMs) [19]. Kyoung-jae Kim clarifies the feasibility of SVM application by comparing with a back propagation neural network. Similarly, for seasonal and trend time series forecasting, a neural network has been widely used. In the paper, Peter Zhang elaborately examines ANN and ARIMA model effectiveness in business and economic sectors [20]. A quasi-linear auto regressive model, vector regressive model, ANN, and ARIMA based model are also used in a similar manner [21–24]. Weather forecasting has been a concern for its low accuracy. In most cases, linear relations between input and output weather data were established, which is primarily incorrect. Abhishek et al. aim to establish a nonlinear relation for weather analysis [25]. The authors improved a statistical model based of ANN which also indicates the behavior of an increased hidden layer on accuracy and performance. Several works related to energy forecasting are also reported. Solar energy forecasting based on a hybrid neural network is proposed in [26]. Similarly, wind power forecasting and multi-step wind speed forecasting are discussed in [27,28], respectively. These statistical models establish a mathematical relation between time series data but lack potential features such as data-mining and feature extracting. For some time series forecasting, it is not feasible to use statistical models due to the intermittent and chaotic nature of those time series like renewable energy forecasting [29]. Deep learning-

based models are ideal for such time series forecasting. An artificial intelligence-based model such as deep leaning can discover abstract feature and invariant structure in data. Consequently, deep leaning is practiced in probabilistic wind power forecasting with high-accuracy and consistency [30]. In [31], a multi-modality graph neural network is proposed addressing the major issues in price prediction in the financial industry. The proposed model was used to learn the lead-lag effects of financial time series. Six deep learning models were employed in [32] for forecasting of cases and death rate for COVID-19 in Australia and Iran. These models include LSTM, Convolutional LSTM, GRU, and their bidirectional extensions. It was found that the bidirectional extension models perform better than the non-bidirectional models. An architecture of LSTM involving Conv1d is proposed in [33] to forecast Web Traffic.

2.2. Forecasting Based Anomaly Detection

Forecasting towards cyber data injection attack detection is a comparatively unexplored region. A comprehensive literature review is done in [34] on an FDI attack. The world is gradually depending more and more on cyber networks. Cyber security is vital because it encompasses everything that pertains to protecting our sensitive data, personally identifiable information, protected health information, personal information, etc. Among a few works, Alfeld et al. elaborated a mathematical framework against a data poisoning attack based on a linear autoregressive model [35]. A similar approach was taken in [36]. In their work, the authors developed an algorithm against anomalies in a power load system. The algorithm, Machine Learning based Anomaly Detection (MLAD), uses neural networks to reconstruct the benchmark and scaling data. Bayes classification, which is based on the cumulative distribution function, is used to estimate a cyber attack template. The authors also prosecute dynamic programming and numerical simulation to calculate load parameters. For smart grid security, a handful of models had been proposed. For example, the vector auto-regressive model and dynamic factor model show promising results against FDI detection [37]. Different approaches were also taken like the use of nonlinear models [38]. Another approach was taken in temperature anomaly detection for electric load [39]. In their work, the authors enhance the accuracy of the final load forecasts by removing the fabricated altered data from the original input data. In supply chain management, a huge amount of information can lead to an overload of data and generate unexpected patterns or anomalies. A machine learning based method such as LSTM and LSTM Auto-encoder techniques can optimally detect anomalies [40]. An LSTM variational Auto-Encoder was implemented in [41] for detecting anomolies in water treatment facilities and water distribution facilities. In [42], several prediction models including LSTM Auto-Encoders were investigated for anomaly detection on a time series of Quebec Metro ridership. In [43], a CNN-LSTM framework was established for anomaly detection for solar power forecasting under cyber-threat. A deep neural network was used in [44] for state prediction and detection of FDI attacks. Most of these detection methods are vulnerable to contamination attacks, FDI attack, or SQL injection. Those models are generalized using an ideal condition as a consequence of a falsified data-set can make the entire system malfunction as well as cause detection delay. On the contrary, in this paper, we took a different approach to sort out detection delay.

3. Proposed Method

The proposed method contains two stages: (i) in the first stage, the deep learning model gives a forecast on real-time data, and (ii) in the second stage, the anomaly detection engine detects anomalies based on the forecast. The proposed method contains two stages: (i) in the first stage, the deep learning model gives a forecast on real-time data, and (ii) in the second stage, the anomaly detection engine detects anomalies based on the forecast.

3.1. Energy Consumption Forecasting Using CNN-LSTM Auto-Encoder Sequence to Sequence Architecture

The CNN-LSTM Auto-encoder architecture for forecasting energy consumption consists of a series connection of CNN and LSTM. The Encoder consists of an input layer, CNN layer, and several hidden layers of pooling layers and flatten layers. Equation (1) is the result of the vector from a convolutional layer, y_{ij} is calculated by input vector x_{ij} to the convolutional layer, b represents bias, w is the weight of the kernel, n is the index value of the filter, and α is the activation function:

$$y_{ij} = \alpha(b_j + \sum_{n=1}^{N} w_{m,j} x^0_{i+m-1,j}) \quad (1)$$

CNN reads across the sequence input and automatically learns the salient features. It projects the results into feature maps. A pooling layer is used to combine the output of a neuron cluster in one layer and convert it into a single neuron in the next layer. It causes reduction in the number of parameters and simplifies the feature maps. It also helps adjusting over fitting. In Equation (2) for a max pooling layer, S is the stride and R is the pooling size, which is less than the size of the input$y$.

$$p_{ij} = \max_{r \in R} y_{i \times S + rj} \quad (2)$$

The output of the pooling layer is flattening into one long vector, which is later interpreted by the decoder. The Decoder consists of only LSTM layer and an output layer. An LSTM layer stores information about the characteristic of the sequence extracted through the CNN Encoder. LSTMs are a special kind on the RNN network that consists of a set of gates that controls the output. LSTMs are explicitly designed to avoid the long-term dependency problem. It also shares the same variables across all time steps, which reduces the number of unknowns significantly. LSTM has a strong ability to capture the dynamic features via cycles in the graph. As the structure of LSTM is shown in Figure 1, given input sequence $X = (x_1, x_2, \ldots\ldots x_N)$, the LSTM computes a two vector sequence, $h = (h_1, h_2, \ldots\ldots h_N)$ and $Y = (y_1, y_2, \ldots\ldots y_N)$. LSTM can remove or add information to the cell state by the gates where cell state c_t is the LSTM structure that stores information. LSTM consists of three sigmoid gates (1) one is for forget cell, f_t that removes old information, (2) the second one adds new information in the cell state, and (3) the third one is used to update old cell state, c_{t-1} into a new cell state c_t. Here, σ is the sigmoid function that has an output range [0,1], *tanh* is the hyperbolic tangent function that has an output range $[-1,1]$, and the purple dotted circles represent pointwise operations. Therefore, LSTM was chosen after the CNN layer to learn dependencies in the sequence of time series data.

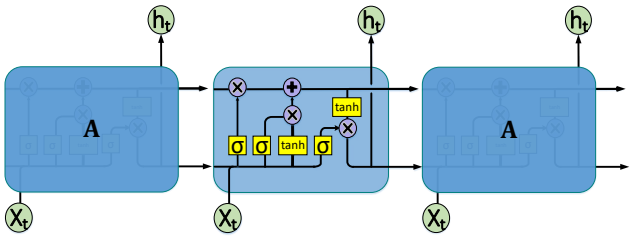


Figure 1. The repeating module in an LSTM containing four interacting layers.

The LSTM transition functions are defined as follows:

$$f_t = \sigma(W_f \times [C_{t-1}, h_{t-1}, x_t] + b_f \quad (3)$$

$$i_t = \sigma(W_f \times [C_{t-1}, h_{t-1}, x_t] + b_i \quad (4)$$

$$C'_t = tanh(W_C \times [h_{t-1}, x_t] + b_C \quad (5)$$

$$o_t = \sigma(W_f \times [C_t, h_{t-1}, x_t] + b_o \quad (6)$$

$$C_t = f_t \times C_t + i_t * C_t^{/} \quad (7)$$

$$h_t = o_t \times tanh(c_t) \quad (8)$$

LSTM learns the temporal relationship on a long-term sequence. It is well suited for forecasting time series. The encoder converts input sequences of variable length into a fixed length vector that is used as the input state for the decoder and the decoder generates an output sequence of l length. In training, back-propagation signals flow from a decoder to encoder network. As a result, weights for the network are updated in order to minimize the following error Equation (9). The whole structure of the deep learning network is shown in Figure 2:

$$L_E = \sum (y_t - \widehat{y}_t)^2 \quad (9)$$

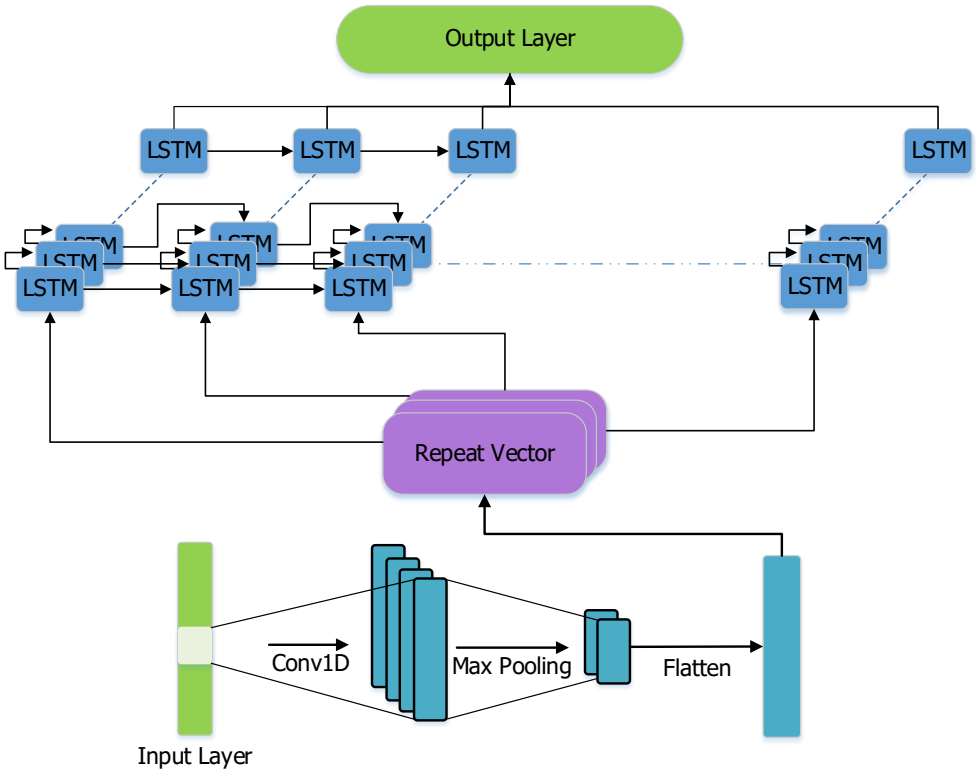


Figure 2. CNN-LSTM auto-encoder sequence to sequence architecture.

3.2. Hyperparameter Tuning

For optimizing the hyperparameters of proposed model, we first listed the different combination of hyperparameters such as the filter number and kernel size of conv1d, number of units in LSTM layer, etc. A training loop was created where all the different combinations of hyperparameters of our forecasting model were implemented, and we found out the Mean Absolute Error (*MAE*) in each combination. From a log of MAE of each model, we obtained the optimal one. *MAE* is described in Section 4.2. The process is summed up in Figure 3. Table 1 shows the optimal set of hyperparameters of the model.

Table 1. Optimized hyperparameters of the proposed model.

Hyperparameter Type	Hyperparameter Value
Conv1d Filter number	32
Conv1d Kernel size	3
Maxpooling1d pool size	2
LSTM unit number	220

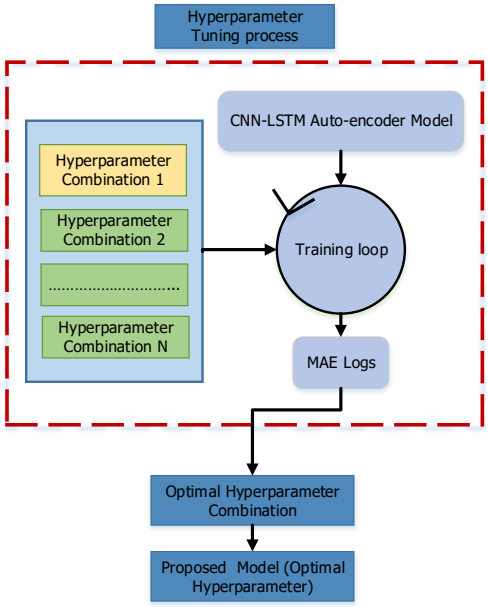


Figure 3. Hyperparameter tuning.

3.3. Real-Time Anomaly Detection Based on Accurate Forecasting

The anomaly detection algorithm attempts to find abnormal behavior in the collected data. Historical data of previous time periods are stored in SCADA historians, which are used to forecast current time steps with the CNN-LSTM Auto-encoder model. Based on the forecasted data of a certain time period 't', irregularities in the measured real-time data of that period can be detected.

In this step, the algorithm compares the forecasted value of an instance 'i' of time period 't' based on previous time period 't-1' with a real-time measured value. The range $[d_t^{lb}, d_t^{ub}]$ where d_t^{lb} and d_t^{ub} is given by Equations (10) and (11); then, there is no anomaly, and, if it does not, then there is an anomaly in the system shown in Equation (12):

$$d_t^{lb} = \mu - \sigma \times \mu \tag{10}$$

$$d_t^{ub} = \mu + \sigma \times \mu \tag{11}$$

$$d_t = \begin{cases} NoAnomaly & \text{if } d_t^{lb} \leq d_t \leq d_t^{ub} \\ Anomaly & \text{if } d_t < d_t^{lb}, d_t > d_t^{ub} \end{cases} \tag{12}$$

Here, μ is the forested value of an interval 'i' and σ is the threshold parameter, which is empirically determined. The algorithm checks all the instances of in the time period and determines whether or not if there is an anomaly like in Figure 4. The flowchart of the algorithm is given in Figure 5.

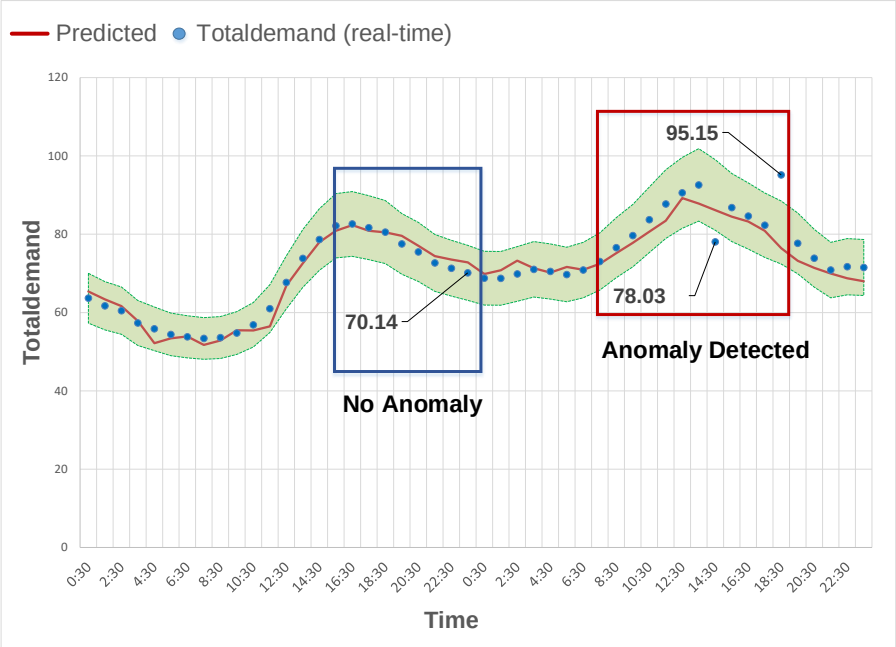


Figure 4. Real-time anomaly detection of whole time period 't'.

The CNN-LSTM Auto-encoder is trained with a sliding approach as shown in Figure 6 [45]. The first few intervals are counted as input, and the forecasting model predicts the energy consumption of the next intervals. Then, it moves next to the window of the same time period and at the same time the anomaly detection tries to find the irregularities of the system by the above-mentioned process.

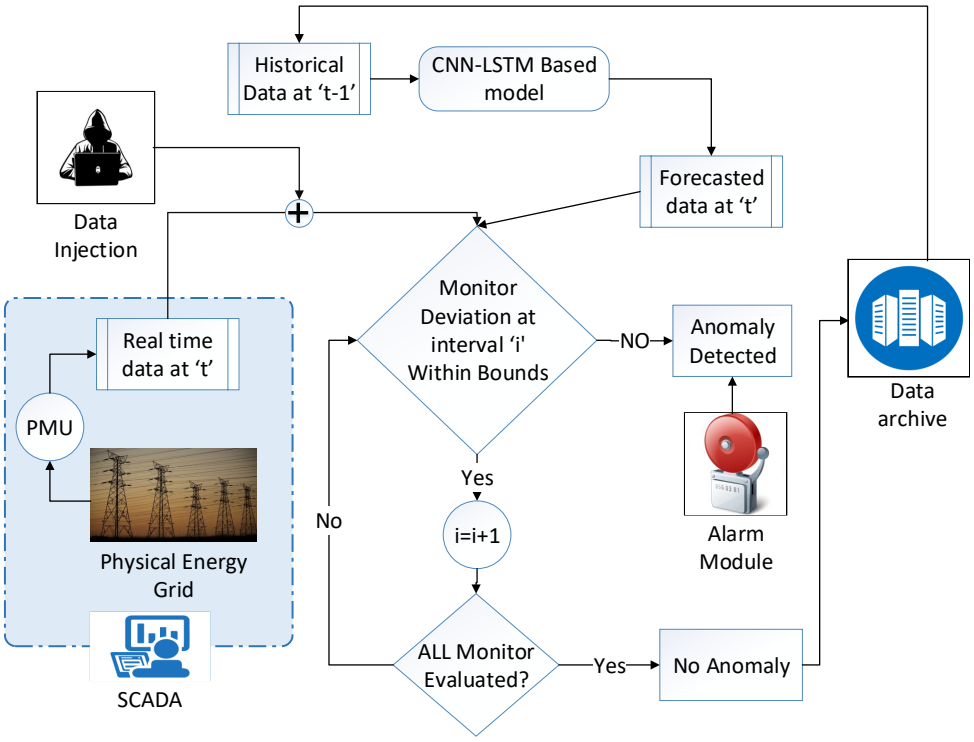


Figure 5. The whole Anomaly Detection Engine with SCADA included.

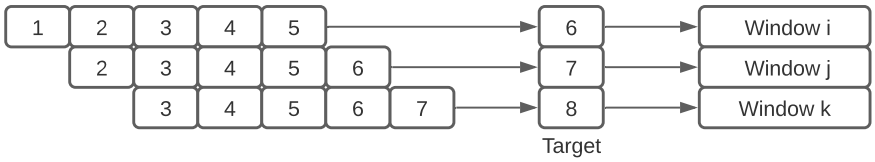


Figure 6. Sliding window.

The heart of the proposed framework is illustrated in Figure 7, where "Data Archived" which contain all historical data plays a critical role. It is assumed that the historical data are attack free which is basically used as a ground truth for training the model towards building the profiles of normal behaviours. Data treatment is vital for training of the CNN-LSTM based model which may contain data sheet preparation, scaling, feature extraction, etc. Gain and allowance for particular time period 't' is the terminal outcome of the trained model. In this framework, raw data input from the energy grid is sensed by the Phasor Measurement Units (PMU) units and then relocated to SCADA; any type of injection attack may materialize concurrently. Real-time data and Forecasted data are then fed to the logic loop that inspect for irregularities, disturbance, and contortion of features. Anomaly is recognized when any type of disturbance is detected. Any type of disturbance within a bound is assumed to be a negligible system error. If anomaly is perceived in any loop of interval 'i' a alarm is raised, and the system is brought to a halt; if not, PMU reading is considered as interruption free and stored in the "Data archive" for use of the next loop. This process reiterates for the next time period.

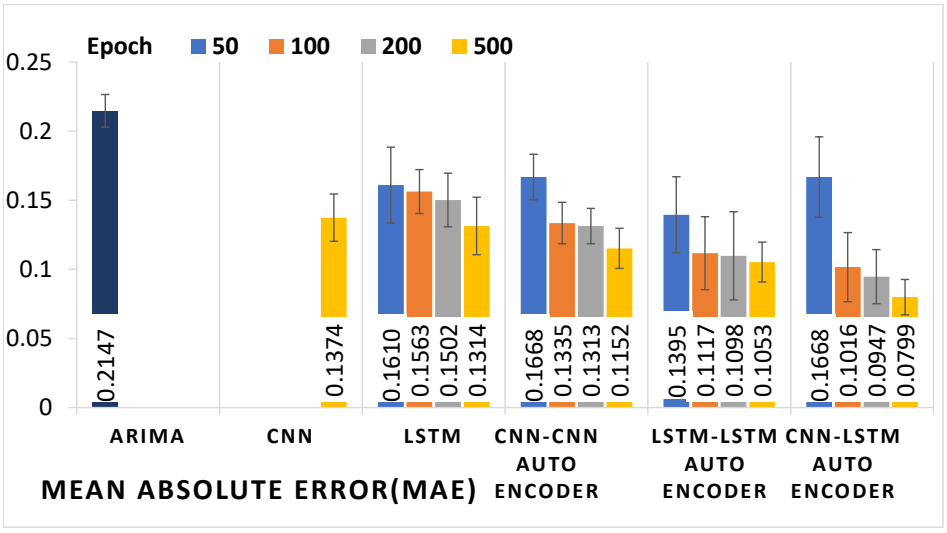


Figure 7. *MAE* of different forecasting models.

4. Results and Evaluation

The performance of proposed method is evaluated on the basis of a False Positive (*FP*) rate, False Negative (*FN*) rate, and Accuracy (*ACC*) Value. The *FP* rate, *FN* rate, and *ACC* value vary with the variation of σ value of the proposed method. A test dataset was created to find out the False Positive (*FP*) rate, False Negative (*FN*) rate, and Accuracy (*ACC*) Value of the proposed method.

4.1. Experimental Dataset

The dataset consisting of three months of data on energy demand consumption was taken from Australian Energy Market Operator (AEMO) [46]. In addition, 65% of the dataset was used for training the model and rest was used for testing.

4.2. Error Matrix

Several error measures are used to evaluate the forecasting models, the Mean Absolute Error (*MAE*) as shown in Equation (13), Mean Absolute Percentage Error (*MAPE*) shown in Equation (14), Mean Squared Error (*MSE*) shown in Equation (15), and Root Mean Squared Error (*RMSE*) as shown in Equation (16). Table 2 shows the comparison between the forecasting models in all the mentioned error matrixes. *MAE* evaluates the absolute deviation between the measured energy consumption and predicted energy consumption. It also takes into account the negative errors. *MAPE* is a statistical measure which is also used to explicate accuracy of forecasting models. Another approach is to square the deviations, which is done in *MSE*. This weighs the greater deviations more than the lower one. However, the units are squared, which could be unwanted. To fix this, *RMSE* is used, which returns the original unit. All these error matrixes together give a clear picture about the accuracy of forecasting models. The equations use predicted energy consumption value '$\widehat{Y}$' and measured energy consumption value 'Y'. The number of samples are 'N':

$$MAE = \frac{1}{N}\sum|\widehat{Y} - Y| \tag{13}$$

$$MAPE = \frac{1}{N}\sum|\frac{\widehat{Y} - Y}{Y}| \times 100\% \tag{14}$$

$$MSE = \frac{1}{N}\sum(\widehat{Y} - Y)^2 \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum(\widehat{Y} - Y)^2} \tag{16}$$

Table 2. Comparison among different forecasting models.

Models	Epoch	*MAE*	*MAPE* (%)	*MSE*	*RMSE*
ARIMA	-	0.2147 ± 0.0118	32.68 ± 2.69	0.0682 ± 0.0014	0.2612 ± 0.0339
CNN	500	0.1374 ± 0.0171	22.98 ± 0.86	0.0249 ± 0.0003	0.1581 ± 0.0181
LSTM	50	0.1610 ± 0.0274	27.24 ± 1.32	0.0370 ± 0.0004	0.1924 ± 0.0209
	100	0.1563 ± 0.0159	24.39 ± 0.85	0.0369 ± 0.0006	0.1922 ± 0.0249
	200	0.1502 ± 0.0194	23.09 ± 0.92	0.0338 ± 0.0006	0.1840 ± 0.0257
	500	0.1314 ± 0.0208	21.52 ± 0.97	0.2461 ± 0.0005	0.1569 ± 0.0238
CNN-CNN	50	0.1668 ± 0.0165	26.15 ± 0.87	0.0355 ± 0.0002	0.1886 ± 0.0165
Auto-encoder	100	0.1335 ± 0.0150	22.18 ± 0.72	0.0246 ± 0.0003	0.1570 ± 0.0179
	200	0.1313 ± 0.0128	21.69 ± 0.65	0.0240 ± 0.0003	0.1552 ± 0.0174
	500	0.1152 ± 0.0145	18.08 ± 0.69	0.0186 ± 0.0002	0.1366 ± 0.0159
LSTM-LSTM	50	0.1395 ± 0.0275	22.49 ± 2.86	0.0334 ± 0.0013	0.1830 ± 0.0362
Auto-encoder	100	0.1117 ± 0.0264	17.65 ± 1.16	0.0246 ± 0.0019	0.1570 ± 0.0443
	200	0.1098 ± 0.0319	13.07 ± 1.82	0.0201 ± 0.0016	0.1420 ± 0.0412
	500	0.1053 ± 0.0144	12.09 ± 0.74	0.0184 ± 0.0003	0.1357 ± 0.0175
CNN-LSTM	50	0.1668 ± 0.0291	25.91 ± 2.95	0.0357 ± 0.0012	0.1886 ± 0.0347
Auto-encoder	100	0.1016 ± 0.0250	15.96 ± 1.45	0.0170 ± 0.0002	0.1305 ± 0.0165
	200	0.0947 ± 0.0196	10.87 ± 1.22	0.0148 ± 0.0007	0.1217 ± 0.0281
	500	0.0799 ± 0.0128	8.07 ± 0.62	0.0106 ± 0.0003	0.1033 ± 0.0176

Bar charts of *RMSE* and *MAE* of the models are shown, in Figures 7 and 8. Table 2, Figures 7 and 8 are discussed in Section 4.7.

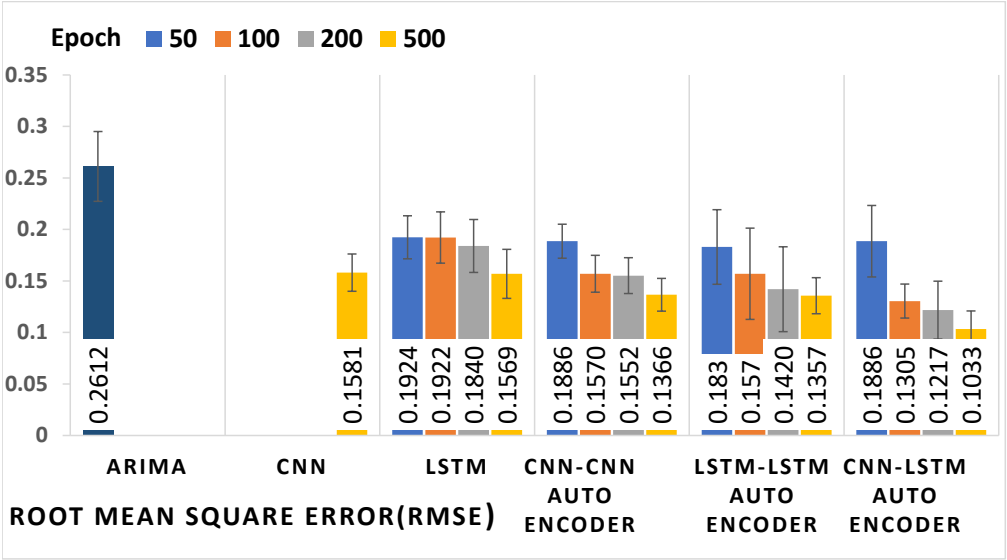


Figure 8. *RSME* of different forecasting models.

4.3. Results

The performance of the proposed method is evaluated on the basis of False Positive (*FP*) rate, False Negative (*FN*) rate, and Accuracy(*ACC*) Value. The *FP* rate, *FN* rate, and *ACC* value vary with the variation of σ value of the proposed method. A test dataset was created to find out the False Positive *FP* rate, False Negative *FN* rate, and Accuracy *ACC* Value of the proposed method. Thus, to obtain the highest performance from the Anomaly Detection Engine, we have to find the optimum σ value of the method, which yields low *FP* rate, low *FN* rate, and high *ACC* value.

Three types of attack scenarios were included in the the test dataset. These attack scenarios are modeled by particular adversary models, which are described in [47].

Attack Types

In a scaling attack, true measurements are multiplied by a scaling parameter γ_S to a higher or lower value for a time period. It is given by

$$P_t^F = (1 + \gamma_S) \times P_t, \ for \ t_s < t < t_e \tag{17}$$

Here, P_t is the measured true value, t_s is the starting time and t_e is the ending time for the cyber-attack duration, and P_t^F is the modified value after cyber-attack.

In a ramp attack, true measurements are modified in manner where the value increases or decreases for a time period by a ramping parameter γ_R. There are two types of Ramp attack. In Type I, only Up-ramping is considered, and, in Type II, both Up-ramping and Down-ramping are included. Type I is given by Equation (18), and Type II is given by Equations (19) and (20):

$$P_t^F = \gamma_R \times (t - ts) \times P_t, \ for \ t_s < t < t_e \tag{18}$$

$$P_t^F = [1 + \gamma_R \times (t - ts)] \times P_t, \ for \ t_s < t < (t_s + t_e)/2 \tag{19}$$

$$P_t^F = [1 + \gamma_R \times (t - ts)] \times P_t, \ for \ (t_s + t_e)/2 < t < t_e \tag{20}$$

Here, t_s is the starting time, and t_e is the ending time for the cyber-attack duration.

Pulse attacks are opposite to scaling attack in kind where the true measurements are modified to higher or lower values for a specified short time period by a parameter γ_P. It is given as below:

$$P_t^F = (1 + \gamma_P) \times P_t, \ for \ t = t_P \tag{21}$$

Here, t_P is the occurrence time of the cyber-attack.

These attacks were randomly distributed, and the attack magnitude was scaled from 2 to 25% of the measured true value.

4.4. False Positive Rate

False Positive is the case where the algorithm accounts a true measurement value as an irregularity. If the False Positive rate is excessively high, many of the true values will be accounted as anomalies, and there will be less confidence in the framework. *FP* Rate is calculated from Equation (22), where *TN* refers to True Negative. A 0% False Positive rate means that the proposed method can classify every secure measurement value as secure:

$$FPRate = \frac{FPCounts}{FPCounts + TNCounts} \tag{22}$$

Figure 9 shows the variation of a false negative rate for different values of σ. With the increase of σ, the False positive Rate decreases. On the x-axis, the σ value is given from 0.00 to 0.12, and the y-axis indicates percentage of False Positive or False Negative. For the σ value beyond 0.105, the False positive becomes zero. Below the 0.06 σ value, the *FP* Rate is significantly high.

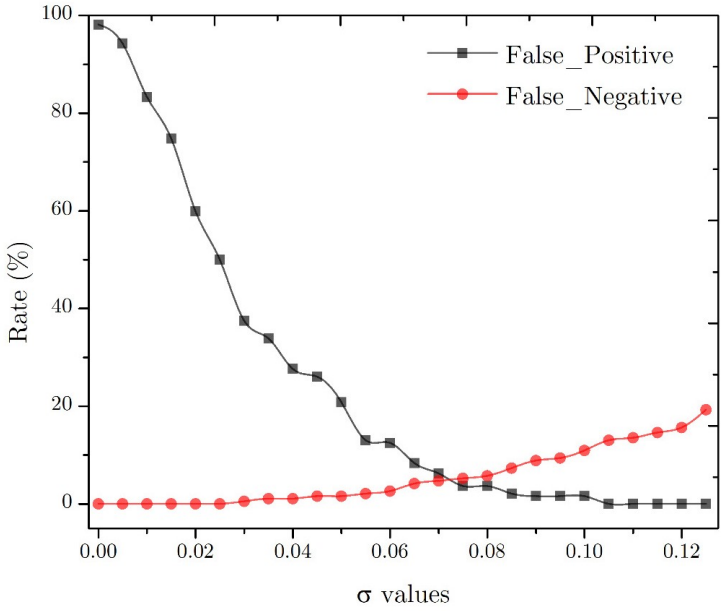


Figure 9. Variation of false negative rate and false positive rate for different values of σ.

4.5. False Negative Rate

False Negative is the case where the algorithm wrongly classifies an abnormality as a true value. A high False Negative rate implies that the algorithm fails to detect many cases of anomalous behavior in the system. *FN* Rate is calculated from Equation (23), where *TN* refers to True Positive. A 0% False Negative rate means the Anomaly Detection Algorithm can detect every attack:

$$FNRate = \frac{FNCounts}{FNCounts + TPCounts} \tag{23}$$

Figure 9 presents the variation of *FN* Rate for different values of σ value. On the x-axis, the σ value is given from 0.00 to 0.12, and the y-axis indicates a percentage of False Positive or False Negative. For σ value above 0.09, the *FN* Rate becomes significantly high, and for σ value below 0.025, the *FN* Rate is zero. Above the 0.095 σ value, the False Negative rate begins to have significantly high value.

4.6. Accuracy Value

Accuracy value can be defined as how many cases where the anomaly detection algorithm can classify an attack as an attack and a secure measurement value as secure. Accuracy Value is calculated from Equation (24). A 100% accuracy value means that the proposed method can detect every attack and also classify every secure measurement value. It is as follows:

$$ACC = \frac{TPCounts + TNCounts}{FPCounts + TPCounts + TNCounts + FNCounts} \quad (24)$$

Here, *ACC* is accuracy value. *TP*, *TN*, *FP*, and *FN* are True Positive Counts, True Negative Counts, False Positive Counts and False Negative Counts. Figure 10 presents the variation of *ACC* value for σ values from 0.05 to 0.12. On the x-axis, the σ value is given from 0.00 to 0.12, and the y-axis indicates Accuracy Value from 88% to 96%. At .075 σ value, the *ACC* value is 95.58%, which is the highest. From 0.075 to 0.09 σ value, the *ACC* value is above 95%. After 0.09 σ value, the *ACC* value begins to drop.

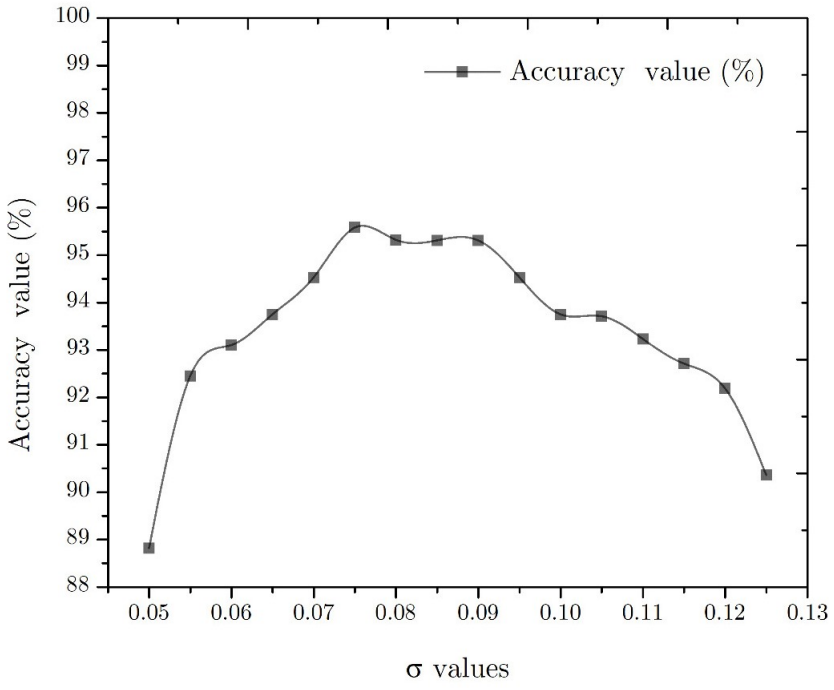


Figure 10. Variation of accuracy value for different values of σ.

4.7. Discussion

A comparison was made between different forecasting models like statistical model ARIMA, Deep Learning models like LSTM, CNN, CNN-CNN Auto-encoder, LSTM-LSTM Auto-encoder, and CNN-LSTM Auto-encoder. In Table 2, *MAE*, *MAPE*, *MSE* and *RMSE* of the mentioned models are shown. The first column represents the forecasting models, the second column holds the Epoch number, the third, the fourth, the fifth and the sixth ones hold *MAE* and *MAPE*,*MSE*,*RMSE*, respectively. The color coding gives a good overview with a deep red color indicating large error and a deep green color indicating small error. Table 2 shows that statistical model ARIMA has larger *MAE* and *RMSE* than all the deep learning models. However, all deep learning models trained with a small epoch number also have large *MAE* and *RMSE*. As the models are trained with more epoch numbers, the *MAE* and *RMSE* become smaller and smaller. However with larger number of epochs, the models require more time and, up to a certain number of epochs, the *MAE* and *RMSE* remain more or less the same. Among all the deep learning models, training the LSTM-LSTM Auto-encoder requires more time than others. The CNN-LSTM Auto-encoder with

500 epochs has the smallest *MAE*, *MAPE*, *MSE* and *RMSE* among all the mentioned models. The CNN-LSTM Auto-encoder outperforms all the other models. The *MAE* and *RMSE* of CNN-LSTM Auto-encoder of 500 epochs are 0.0799 $\pm$ 0.0128 and 0.1033 $\pm$ 0.0176, respectfully. In Figures 7 and 8, bar charts of *RMSE* and *MAE* of the models are shown. The top one shows the *MAE*, and the bottom one shows *RMSE*. Here, the blue bar, orange bar, green bar and yellow bar indicate 50, 100, 200 and 500 epochs, respectfully.

The models do not only have to have a small *MAE* and *RMSE*, but also have to perform well over a long period of time as the effectiveness of the anomaly detection algorithm depends on it. Otherwise, the Anomaly Detection Algorithm will utterly fail to detect anomalies during some period of time, and other times it will give a false alarm. In Figures 7 and 8, a comparison is made among all the mentioned models over a sample time period. The *x*-axis indicates time in hour:minute and the *y*-axis indicates total power demand. Here, the black line is actual measurement of energy consumption, and the purple line is the predictions of CNN-LSTM Auto-encoder model in Figure 11.

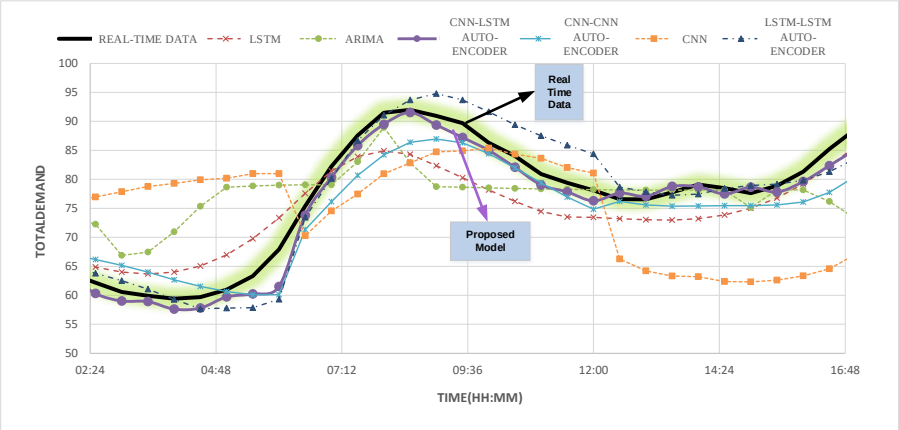


Figure 11. The performance of different models in a time period.

Here, we can see that the predictions are relatively close to the actual measurement of energy consumption value. In Figure 12, the performance of the CNN-LSTM Auto-encoder in two days is shown. Here, the *x*-axis also indicates time in hour:minute and the *y*-axis indicates total power demand. Here, the allowance is 10% of the measured value in every instance. It is shown in Figure 12 that the CNN-LSTM Auto-encoder is performing consistently well for two-day time periods and is under 10% margin of the actual measurement of energy consumption value.

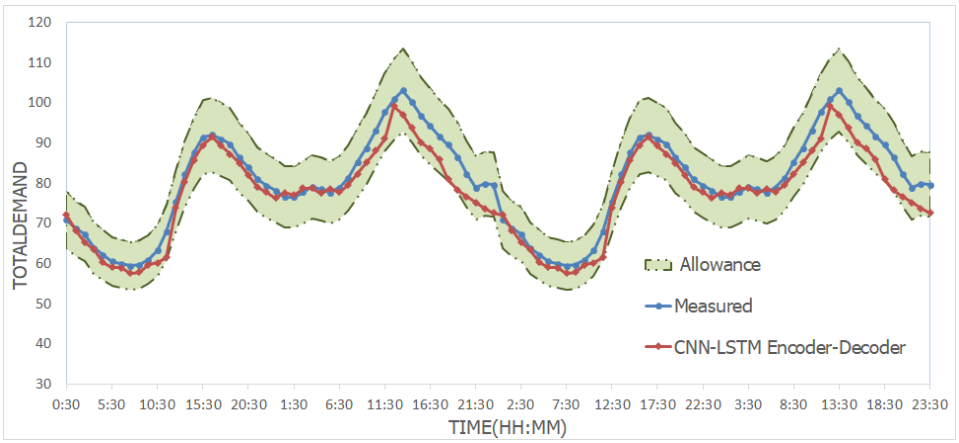


Figure 12. The performance of the CNN-LSTM Auto-encoder in a two-day period.

5. Conclusions

In this paper, a multi-stage anomaly detection framework is developed to address the data integrity issues of the meter data management. The Proposed Method has high accuracy, a low false positive rate and a low false negative rate. However, the accuracy of the forecasting model is essential for the anomaly detection engine to perform well. Thus, the forecasting result of the CNN-LSTM Auto-encoder model is a very critical step for the anomaly detection engine to perform properly; otherwise, it will have a high false positive rate. In this paper, the CNN-LSTM Auto-encoder model performs significantly better compared to other benchmark algorithms in terms of forecasting accuracy considering the real-world dataset from AEMO. The proposed model with better forecasting identifies anomalies with high detection accuracy with low false positives and shows its effectiveness towards PMU data integrity.

Author Contributions: Conceptualization, A.A.; methodology, A.M.-a.-r., F.H. and A.A.; software, A.M.-a.-r. and F.H.; validation, A.A. and S.A.; formal analysis, A.M.-a.-r. and F.H.; investigation, A.M.-a.-r. and F.H.; resources, A.A. and S.A.; data curation, A.M.-a.-r.; writing—original draft preparation, A.M.-a.-r. and F.H.; writing—review and editing, A.A. and S.A.; visualization, A.M.-a.-r. and F.H.; supervision, A.A.; project administration, A.A.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SCADA	Supervisory Control and Data Acquisition
FDI	False Data Injection
PMU	Phasor Measurement Units
CNN	Convolutional Neural Network
LSTM	Long Short-term Memory
SVM	Support Vector Machine
ANN	Artificial Neural Networks
ARIMA	Autoregressive Integrated Moving Average
MLAD	Machine Learning based Anomaly Detection
FP	False Positive
FN	False Negative
ACC	Accuracy
AEMO	Australian Energy Market Operator
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error

References

1. Güngör, V.C.; Sahin, D.; Kocak, T.; Ergüt, S.; Buccella, C.; Cecati, C.; Hancke, G.P. Smart grid technologies: Communication technologies and standards. *IEEE Trans. Ind. Inform.* **2011**, *7*, 529–539. [CrossRef]
2. Wang, Y.; Pordanjani, I.R.; Xu, W. An event-driven demand response scheme for power system security enhancement. *IEEE Trans. Smart Grid* **2011**, *2*, 11–17. [CrossRef]
3. Anwar, A.; Mahmood, A.N.; Pickering, M. Modeling and performance evaluation of stealthy false data injection attacks on smart grid in the presence of corrupted measurements. *J. Comput. Syst. Sci.* **2017**, *83*, 58–72. [CrossRef]

4. Liang, G.; Weller, S.R.; Zhao, J.; Luo, F.; Dong, Z.Y. The 2015 Ukraine Blackout: Implications for False Data Injection Attacks. *IEEE Trans. Power Syst.* **2017**, *32*, 3317–3318. [CrossRef]
5. Mallick, P. *Cyber Attack on Kudankulam Nuclear Power Plant—A Wake Up Call*; Vivekananda International Foundation: New Delhi, India, 2019.
6. Lab, K. Threat Landscape for Industrial Automation Systems in H2 2019. *ICS Cert* **2019**, 1–37.
7. Morris, T.; Pan, S.; Lewis, J.; Moorhead, J.; Younan, N.; King, R.; Freund, M.; Madani, V. Cybersecurity risk testing of substation phasor measurement units and phasor data concentrators. In Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, Oak Ridg, TN, USA, 12–14 October 2011. [CrossRef]
8. Dondossola, G.; Szanto, J.; Masera, M.; Fovino, I.N. Effects of intentional threats to power substation control systems. *Int. J. Crit. Infrastructures* **2008**, *4*, 129–143. [CrossRef]
9. Wang, J.W.; Rong, L.L. Cascade-based attack vulnerability on the US power grid. *Saf. Sci.* **2009**, *47*, 1332–1336. [CrossRef]
10. Huang, Z.; Wang, C.; Zhu, T.; Nayak, A. Cascading failures in smart grid: Joint effect of load propagation and interdependence. *IEEE Access* **2015**, *3*, 2520–2530. [CrossRef]
11. LaWell, M. The state of industrial: Robots. *Ind. Week* **2017**, *266*, 10–13.
12. Stouffer, K.; Falco, J.; Scarfone, K. GUIDE to industrial control systems (ICS) security. *Stuxnet Comput. Worm Ind. Control Syst. Secur.* **2011**, *800*, 11–158.
13. Anwar, A.; Mahmood, A.N. Vulnerabilities of Smart Grid State Estimation Against False Data Injection Attack. In *Green Energy and Technology*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 411–428. [CrossRef]
14. Inl/ext-06; Kuipers, D.; Fabro, M. *Control Systems Cyber Security: Defense in Depth Strategies*; Idaho National Laboratory: Idaho Falls, ID, USA, 2006; p. 8.
15. Ameli, A.; Hooshyar, A.; El-saadany, E.F.; Youssef, A.M.; Member, S. Generation Control Systems. Attack Detection and Identification for Automatic Generation Control Systems. *IEEE Trans. Power Syst.* **2018**, *33*, 4760–4774. [CrossRef]
16. Musleh, A.S.; Chen, G.; Dong, Z.Y. A Survey on the Detection Algorithms for False Data Injection Attacks in Smart Grids. *IEEE Trans. Smart Grid* **2020**, *11*, 2218–2234. [CrossRef]
17. Huang, Y.F.; Werner, S.; Huang, J.; Kashyap, N.; Gupta, V. State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid. *IEEE Signal Process. Mag.* **2012**, *29*, 33–43. [CrossRef]
18. College, D.S.; Helms, M.M.; Chapman, S. supply chain management Supply chain forecasting. *Bus. Process Manag. J.* **2000**, *6*, 392–407.
19. Kim, K.J. Financial time series forecasting using support vector machines. *Neurocomputing* **2003**, *55*, 307–319. [CrossRef]
20. Zhang, G.P.; Qi, M. Neural network forecasting for seasonal and trend time series. *Eur. J. Oper. Res.* **2005**, *160*, 501–514. [CrossRef]
21. Gan, M.; Cheng, Y.; Liu, K.; Zhang, G.L. Seasonal and trend time series forecasting based on a quasi-linear autoregressive model. *Appl. Soft Comput. J.* **2014**, *24*, 13–18. [CrossRef]
22. Pai, P.F.; Lin, K.P.; Lin, C.S.; Chang, P.T. Time series forecasting by a seasonal support vector regression model. *Expert Syst. Appl.* **2010**, *37*, 4261–4265. [CrossRef]
23. Khashei, M.; Bijari, M.; Hejazi, S.R. Combining seasonal ARIMA models with computational intelligence techniques for time series forecasting. *Soft Comput.* **2012**, *16*, 1091–1105. [CrossRef]
24. Chiemeke, S.C.; Oladipupo, A.O. African Journal of Science and Technology (AJST). *Engineering* **1982**, *2*, 101–107.
25. Abhishek, K.; Singh, M.; Ghosh, S.; Anand, A. Weather Forecasting Model using Artificial Neural Network. *Procedia Technol.* **2012**, *4*, 311–318. [CrossRef]
26. Abedinia, O.; Amjady, N.; Ghadimi, N. Solar energy forecasting based on hybrid neural network and improved metaheuristic algorithm. *Comput. Intell.* **2018**, *34*, 241–260. [CrossRef]
27. Han, L.; Romero, C.E.; Yao, Z. Wind power forecasting based on principle component phase space reconstruction. *Renew. Energy* **2015**, *81*, 737–744. [CrossRef]
28. Ahmed, A.; Khalid, M. An intelligent framework for short-term multi-step wind speed forecasting based on Functional Networks. *Appl. Energy* **2018**, *225*, 902–911. [CrossRef]
29. Wang, H.; Lei, Z.; Zhang, X.; Zhou, B.; Peng, J. A review of deep learning for renewable energy forecasting. *Energy Convers. Manag.* **2019**, *198*, 111799. [CrossRef]
30. Wang, H.Z.; Li, G.Q.; Wang, G.B.; Peng, J.C.; Jiang, H.; Liu, Y.T. Deep learning based ensemble approach for probabilistic wind power forecasting. *Appl. Energy* **2017**, *188*, 56–70. [CrossRef]
31. Cheng, D.; Yang, F.; Xiang, S.; Liu, J. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognit.* **2022**, *121*, 108218. [CrossRef]
32. Ayoobi, N.; Sharifrazi, D.; Alizadehsani, R.; Shoeibi, A.; Gorriz, J.M.; Moosaei, H.; Khosravi, A.; Nahavandi, S.; Gholamzadeh Chofreh, A.; Goni, F.A.; et al. Time series forecasting of new cases and new deaths rate for COVID-19 using deep learning methods. *Results Phys.* **2021**, *27*, 104495. [CrossRef]
33. Casado-Vara, R.; Martin del Rey, A.; Pérez-Palau, D.; De-la Fuente-Valentín, L.; Corchado, J.M. Web Traffic Time Series Forecasting Using LSTM Neural Networks with Distributed Asynchronous Training. *Mathematics* **2021**, *9*, 421. [CrossRef]
34. Reda, H.T.; Anwar, A.; Mahmood, A. Comprehensive survey and taxonomies of false data injection attacks in smart grids: Attack models, targets, and impacts. *Renew. Sustain. Energy Rev.* **2022**, *163*, 112423. [CrossRef]

35. Alfeld, S.; Zhu, X.; Barford, P. Data poisoning attacks against autoregressive models. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016, Phoenix, AZ, USA, 12–17 February 2016; pp. 1452–1458.
36. Cui, M.; Wang, J.; Yue, M. Machine Learning-Based Anomaly Detection for Load Forecasting Under Cyberattacks. *IEEE Trans. Smart Grid* **2019**, *10*, 5724–5734. [CrossRef]
37. Drayer, E.; Routtenberg, T. Detection of false data injection attacks in smart grids based on graph signal processing. *IEEE Syst. J.* **2019**, *14*, 1886–1896. [CrossRef]
38. Zhao, J.; Zhang, G.; Dong, Z.Y.; Wong, K.P. Forecasting-aided imperfect false data injection attacks against power system nonlinear state estimation. *IEEE Trans. Smart Grid* **2016**, *7*, 6–8. [CrossRef]
39. Sobhani, M.; Hong, T.; Martin, C. Temperature anomaly detection for electric load forecasting. *Int. J. Forecast.* **2020**, *36*, 324–333. [CrossRef]
40. Nguyen, H.D.; Tran, K.P.; Thomassey, S.; Hamad, M. Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. *Int. J. Inf. Manag.* **2020**, *57*, 102282. [CrossRef]
41. Fährmann, D.; Damer, N.; Kirchbuchner, F.; Kuijper, A. Lightweight Long Short-Term Memory Variational Auto-Encoder for Multivariate Time Series Anomaly Detection in Industrial Control Systems. *Sensors* **2022**, *22*, 2886. [CrossRef]
42. Pasini, K.; Khouadjia, M.; Samé, A.; Trépanier, M.; Oukhellou, L. Contextual anomaly detection on time series: A case study of metro ridership analysis. *Neural Comput. Appl.* **2022**, *34*, 1483–1507. [CrossRef]
43. Sun, M.; He, L.; Zhang, J. Deep learning-based probabilistic anomaly detection for solar forecasting under cyberattacks. *Int. J. Electr. Power Energy Syst.* **2022**, *137*, 107752. [CrossRef]
44. Reda, H.T.; Anwar, A.; Mahmood, A.; Chilamkurti, N. Data-driven Approach for State Prediction and Detection of False Data Injection Attacks in Smart Grid. *J. Mod. Power Syst. Clean Energy* **2022**, 1–13. [CrossRef]
45. El Hariri, M.; Harmon, E.; Youssef, T.; Saleh, M.; Habib, H.; Mohammed, O. The IEC 61850 Sampled Measured Values Protocol: Analysis, Threat Identification, and Feasibility of Using NN Forecasters to Detect Spoofed Packets. *Energies* **2019**, *12*, 3731. [CrossRef]
46. Australian Energy Market Operator. Available online: https://aemo.com.au (accessed on 15 February 2022).
47. Sridhar, S.; Govindarasu, M. Model-based attack detection and mitigation for automatic generation control. *IEEE Trans. Smart Grid* **2014**, *5*, 580–591. [CrossRef]