TIME SERIES FORECASTING USING SRL BASED DEEP LEARNING IN SMART GRID

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Taranveer Singh
(Roll No. 801732056)

Under the supervision of:
Dr. Neeraj Kumar
Associate Professor
Computer Science & Engineering Department
Thapar Institute of Engineering & Technology, Patiala

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

June 2019

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"***Time Series Forecasting using SRL based Deep Learning in Smart Grid***"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Neeraj Kumar* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature: Taranveer Singh

(Taranveer Singh)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Neeraj Kumar)
Associate Professor
CSED, TIET Patiala.

ACKNOWLEDGEMENT

After many hours of experimentation, coding, and collaboration that has gone into this thesis, I would not have been able to complete it all myself.

First, I'd like to thank Dr. Neeraj Kumar for being the type of supervisor, who supervised my work and helped me in finding elegant solutions to the problems I encountered during my thesis work, always with a smile.

I would also like to thank the extraordinary people that I encountered during my two years of master, many of which I can now call friends and that have made me love the time spent together.

Thanks to my family, supporting me no matter what, being there in the rough moments, cheering me for every accomplishment and always waiting for me with a smile when I came back to home.

Taranveer Singh

(801732056)

ABSTRACT

Time-series forecasting is one of the most challenging tasks due to the ubiquitousness of the Time series data. Some examples include Astronomical data, weather data, Electricity usage, stock and exchange rates data collected over time. Whereas, in many applications, the availability of labeled data is quite less either due to the privacy or low rate of generation of data.

As a result of this, the small amount of data leads to low performance and overfitting of the machine learning models. In order to deal with small data and low performance, we implemented a generative model based on statistical relational learning and a two-tier ensemble forecasting model to predict the result based on machine learning and deep learning.

Considering the advent and future of the Internet of Things, we choose the smart grid environment to implement the proposed approach because of the availability of the large benchmark dataset UMass Smart* Dataset - 2017 release of smart homes in a locality by taking the reading of appliances and weather conditions with a sampling rate of 1 minute. Use of this dataset helps us in building a robust model that also gets better insights from the data in order to find out the relationship between the device data and the factors affecting the device data to generate the synthetic data from the existing data.

The proposed scheme also shows a significant improvement over the existing load forecasting methods over short-term and long-term load forecasting models with an accuracy of 95.6%

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYM AND ABBREVIATIONS

- SRL: Statistical Relational Learning
- IoT: Internet of Things
- FCRBM: Factored Condition Restricted Boltzmann Machine
- CNN: Convolution Neural Network
- SVM: Support Vector Machine
- SAPSO: Simulated Annealing and Particle Swarm Optimization
- GLSSVM: Group Method of Data Handling and Least Square Support Vector Machine.
- BP: Back Propagation
- ANN: Artificial Neural Network
- MLP: Multi-Layer Perceptron
- RNN: Recurrent Neural Network
- AUROC: Area Under the Receiver Operating Characteristics
- CSV: Comma Separated Values
- TSV: Tab Separated Values
- LSTM: Long-Short Term Memory
- GPU: Graphical Processing Unit
- TPU: Tensor Processing Unit
- MRF: Markov Random Field
- GAN: Generative Adversarial Network
- SVR: Support Vector Regression
- DNN: Deep Neural Network
- FNN: Feed-forward Neural Network
- ReLU: Rectified Linear Unit
- RBF: Radial Based Function
- MAE: Mean Absolute Error

CHAPTER 1

INTRODUCTION

Forecasting Plays a vital role in real-world applications to predict future states or outcomes. These future states or outcomes can help one to govern the decision making, whether that can be done through predicting the valuation, customer churn, demand, or preventative maintenance costs.

Accurate forecasting is all dependent on large historical data to predict the results using statistical models or machine learning algorithms. These algorithms gather patterns from the data to distinguish one situation from the other one. These patterns can only be recognized if there are a sufficient number of samples to cover all the variations that formulate the result or increase in the number of features that leads to result for the defined time stamp.

1.1 Motivation for Data Forecasting in Smart Grid

With the addition of the Internet of things (IoT), major appliances today requires extra electricity unit. With the inclusion of IoT, ICT, and other smart technologies, there has been a transition from electrical grids to smart grids. So, there is a need to increase adaptability and efficiency in terms of load forecasting. To minimize the energy wastage and maximize the optimization, we need to add a high level of flexibility in the smart grids. In order to increase adaptability, the smart grid should take decisions based on the demands. Hence the ability to load forecasting as per the future needs is imperative [1]. The forecasting of the aggregated demand will depend upon the demand generated by its individual components/buildings. These components/buildings can be old or new.

With this increase in the number of units under the smart grid electricity optimization has been a significant concern. This increase in units also leads to an increase in data, which makes all the previous work on load forecasting insignificant and also makes the prediction of energy forecasting a challenging issue. To mitigate this issue, one needs to create a model that can preprocess/train over a large amount of data and provides us accurate predictions.

Nowadays as the energy usage data is easily available by the advent of smart meters, so to mitigate the energy wastage and optimize the energy usage there is a need of an

adaptive model that can accurately predict the short term as well as long term energy load forecasting. Existing systems can only predict the load forecasting either for short-term or for long-term up to a significant level of accuracy [2-3]. They also use only the usage (total energy usage) parameter to predict the result; weather conditions are not taken into account, which affects the forecasting in a significant manner. Weather and climate conditions' data is also easily available due to the weather satellites for every specific region without any efforts.

So the availability of the benchmark dataset UMass Smart* Dataset - 2017 release of smart homes in a locality with weather conditions associated to it makes our work easier and helps us to proceed towards the problem that we tackled.

1.1.1 Problem Statement

As from the motivation part, we come to know that load forecasting is heavily dependent on the data of the region, and there is a need to develop a system for both short and long-term load forecasting.

To mitigate these issues, we need to develop a generalized system that can forecast the result for short and long-term with better performance than the existing system irrespective of the data size i.e., for small datasets too.

So we need to develop a system that can take a small dataset, get insights from the dataset, and then from these insights generates synthetic data to create a large dataset. And this large dataset serves as the input to the forecasting model which will predict the outcome.

1.1.2 Synthetic Data Generation

Data plays a critical role whenever it comes to automate a system using machine learning. But the availability of data publicly of a particular locality is a primary concern. As ongoing transformation from traditional grids to smart grids lead to less amount of data collection of that region. Or due to the privatization of the electricity sector makes availability of electric data usage publicly due to privacy concern or to generate income by selling data.

But using the dataset from a particular locality will not serve as a solution to create a robust system. And using the small dataset for training the forecasting model lead to low performance and overfitting of the model. The only solution to this problem is to create synthetic data from the existing data. The process of synthetic data

generation is quite typical as we have to start from the large dataset and retrieve its properties without affecting its structure and internal data dependencies. The major task is to find out the inner relationships and creating the new data maintaining these internal relationships.

1.1.3 Load Forecasting

Load forecasting is mostly used by the energy supply companies or power utilities to forecast the demand by the consumer over a given period. This given period can be a short-term (up to few hours), medium-term(up to few weeks) or long-term(up to few years) which will help the provider to maintain the balance between the load-demand gap and availability [22]. Short term load forecasting is useful in predicting the peak hours and based on that make the system reliable and decrease the load on the grid by providing the equivalent supply during the non-peak hours. Whereas the medium-term load forecasting is quite useful in predicting the scenario of load and demand on the working days and non-working days in the commercial building, apartments, etc. And long-term load forecasting is used to predict the economic demands that can be useful in future upgrades for the supply systems. Most of the companies forecast over medium and long-term periods [24,25] as it helps them to perform the extension if required in the factories, manufacturing plants, etc.

For better decision making, accurate load forecasting is economically advantageous. But accuracy of the load forecasting depends on the nature of the data such as time stamp size, the number of features, factors affecting the energy consumption e.g., weather, natural disasters, or other technical faults. And based on the nature of the data usage of an appropriate algorithm lead to prevent the incorrect energy billing [23].

1.2 Goals

On a high level, we wish to implement an automated generative model that can generate the synthetic data of multivariate time series and further this data will serve as input to the forecasting model to predict the load. Our main goal should be measured along the following three dimensions:

1. **Generalizability:** The system should be able to forecast the results irrespective of the size of the dataset (~from 500 rows to unlimited)or type of time series dataset, without any modifications.
2. **Accuracy:** The output generated from the synthetic data should be as realistic as possible, i.e., the synthetic data bears all the properties of the real data and lead us to predict the result as predicted from real data.
3. **Optimized:** The system generated from the model should be optimized on a computational level, i.e., can be used over the CPU based or GPU based architecture. And is easily upgradable as per the machine learning framework in which it is based.

1.3 Thesis Roadmap

The rest of the chapters of the thesis are organized as follows:

- Chapter 2 provides the Literature Review and background work done by the researchers.
- Chapter 3 then provides a brief overview of technical concepts and terminologies that are required to understand the experimental setup.
- Chapter 4 covers the Experimental setup and implementation details of the system.
- Chapter 5 is about the results of the implemented system and provides a brief overview of evaluation metrics that are used to validate the findings.
- Chapter 6 then summarizes with the conclusion part, our key findings, and contributions.

By the end of this thesis, you will get a better understanding of the generative model for synthetic data generation and a generalized model for load forecasting.

CHAPTER 2

LITERATURE REVIEW

2.1 For Load Forecasting

Existing systems can only predict the load forecasting either for short-term [3-5] or long-term [2,6]. But they didn't take into account the amount of data which is a major issue. With an increase in data, how efficient the model will be, i.e., adaptive concerning the data. The adaptiveness can be in the form of more optimization or less overfitting. The previous models lead to overfitting with respect to a large dataset. Models should have less parameter and less training size. Most of the existing models didn't take into account the weather condition which is a major factor in predicting the energy [2-9] and those who took, didn't make an adaptive model with respect to the amount of data [9-10].

Mocanu, Elena, et al. [1] proposed the system estimating energy consumption for the smart grid using Factored Condition Restricted Boltzmann Machine (FCRBM). The major disadvantage in this model lies in the dataset Individual household electric power consumption" dataset, which doesn't take into account the weather condition. Also, it uses FCRBM, which involves a higher number of parameters, and The forecasting is done with 1 minute, 15 minute, hourly, and weekly resolutions.

Amarasinghe, Kasun, et al. [2] proposed a system which uses CNN for predicting energy load forecasting on "Individual household electric power consumption dataset." This model was built for long-term forecasting but didn't take into account short-term forecasting. The dataset doesn't take into consideration weather conditions. The prediction was made on 60-hour resolution.

Raza, Muhammad Qamar, and Zuhairi Baharudin. [3] proposed a hybrid neural network forecasting model having a three-layered feedforward neural network with backpropagation using SAPSO which is a combination of simulated annealing (SA) and particle swarm optimization (PSO) called SAPSO for only predicting short term demands. It showed a better result than a conventional neural network, and due to fewer parameters can be easily modified.

Kuo, Ping-Huan, and Chiou-Jye Huang [4] proposed a model of CNN with three convolution layers and three pooling layers for short-term load forecasting. But didn't consider the weather data. The overall performance of the CNN-1d is given in average cumulative variation of Root-Mean-Square Error, which is 11.65942 too large.

Ahmad, A. S., et al. [6] proposed a Group Method of Data Handling (GMDH) and Least Square SVM (LSSVM) combined to make a hybrid model known as GLSSVM, to forecast building electrical energy consumption. The data is collected from Johor Tourism Action Council Johor, from January 1999 to December 2008, in Malaysia. Here also the dataset didn't take into account the weather conditions.

Seetha, Hari, and R. Saravanan [5] used a fuzzy version of the neural network, namely Fuzzy back propagation network (Fuzzy BP) for short term electric load prediction. The fuzzified inputs were fed to the system, and crisp valued output was obtained. The author also shows a comparison of Fuzzy Backpropagation Network with MLP based BP network where Fuzzy based BP network performs better than MLP with BP.

Ho, Kun-Long et al. [7] proposed a multilayer neural network with an adaptive learning algorithm is designed for short term load forecasting. The whole model was tested on the Taiwan Power system. Then the user predicts the output on an hourly basis, i.e., short-term forecasting. Paper also presents the comparison of the simple neural network with back propagation versus the adaptive learning based neural network where the proposed adaptive learning based neural network converges much faster than the simple neural network with Backpropagation.

Bakirtzis, A. G., et al. [8] proposed development of an Artificial Neural Network (ANN) based short-term load forecasting model for the Energy Control Center of the Greek Public Power Corporation (PPC). The forecasting was done on a seven-day time span based on the daily load profile. The daily load profiles show better results than the twelve-noon profile, which trained on the 24 hours of load data collected.

Lijesen, D. P., and J. Rosing [9] proposed a forecasting procedure which is based on forecasting the nominal and residual loads and then summing them to obtain the total load forecast. It takes into account the weather information, but the forecasting is based

on an hourly basis, i.e., short-term forecasting. The average RMS error of the model is 2.1 percent per day. The weekly and seasonal forecasting is proposed as future work.

Roberto Buizza et al. [10] proposed an ANN-based model for Load Forecasting With Weather Ensemble Predictions. Load forecasting is based on medium-term load forecasting. Also, the author showed that the weather conditions that are being taken in this scenario uniquely affect the load forecasting as compare to weather forecasting data.

Sapankevych, Nicholas I., and Ravi Sankar [11] presented a survey on time series prediction for using support vector machines (SVM). Autor gives a detailed review of time series problems being solved by the SVM and then also conclude by a comparison of the SVM model with other machine learning models.

As most of the past work is based on either short-term or long-term load forecasting, which is a major drawback. As to predict the load forecasting, it will lead to training both types of models on the same dataset.

2.2 For Synthetic Time Series Data Generation

Data generation is quite tough in the late 90s, but after the proposal of Generative Adversarial Networks by Ian J. Goodfellow [26], lots of efforts were being made by the researchers in the field of data augmentation especially in Image Processing field. But still, there is a need for creating a system that can generate the synthetic data which can be useful in many fields as most of the field uses the statistical data. Considering the demand now focus is also drawn in this area too, but still, much work is required in this area as to create a model that augments the data which attains maximum properties of the real data.

Anderson, Jason W. et al. [27] proposed a framework for generating synthetic data from IoT. The framework is based on extraction the structure of XML file and then from that structure sustaining the internal properties of data create a synthesis set of values that comprises all the real data properties without compromising the personal privacy concern.

Hu, Joseph W. et al. [28] proposed a framework to create the shadow RDBMS database from the original database. That data is fed into the system as a JSON string

then with algorithm Histogen column-wise data got generated and for further Batchgen and BFS traversal algorithm is used to create the table and maintain the parent-child level relationship.

Myung, Rohyoung, et al. [29] proposed a technique for analysis and generation of the data in the IoT environment. The Data analysis process consists of filtering, discretization, and finding out the dependency and correlation between the data to generate synthetic data. The author also compares to the proposed technique with the Markov chain based model.

Patki, Neha et al. [30] build a system that can create synthetic data from the relational databases. This model is based on generative modeling that uses the statistical parameters in consideration to find out the relationship between the data point. Further, the author tested the synthetic data for training purposes and found out that the generated data can replace the real data. But the given model is only applicable to relational databases. The author suggests the time series data generation in future work.

Esteban, Cristóbal et al. [31] proposed a generative model based on RNN to generate time series data. Data set used in this is ICU data with more than 224 million entries. The basic ideology behind this is to use the memory feature of RNN to predict the relationship of the particular timestamp with the preceding ones. Then further performance is measured by using the random forest classifier trained on real and synthetic data using the AUROC evaluation metric.

CHAPTER 3

TECHNICAL BACKGROUND

To understand the experimental setup in this chapter, we provide the technical aspects of the methodologies used to create the system. This chapter gives you a thorough study of the key concept that made the whole system in a sequential manner as used in the system. The section starts with a brief overview of the system and then proceed with the technical concept and terminologies require to understand the whole scenario.

3.1 Overview

The entire system is divided into two parts. In the first part, the process of synthetic data generation takes place, and in the second part, the forecasting model is used to train over the data and predict the result. The process starts with the multivariate time series data of energy usage of a smart grid with a time stamp of one minute. And then a generative model with statistical relational learning to create synthetic data. After that, the data is used as input for the load forecasting model based on machine learning and deep learning. To get a full understanding, we start with the time series data and then learn Machine learning and deep learning in a theoretical manner. After that, the basic building block of the generative model will extract the feature and relationship from the data, i.e., statistical relational learning concept is discussed. And then the details of the generative models are going to be addressed. Finally, the second phase of the system, i.e., the forecasting models will be discussed that are being used to build the system.

3.2 Concepts and Terminologies

3.2.1 Time Series Data: Time series data as the name signifies the data about something that happened in a series of the time interval. Time series data is a structured data type that consists of rows and column about the event that occurred over a period. Mostly time series data is available in tabular format. But when the entries are quite large, then we represent them either in the excel format or CSV (comma separated file) or tsv (tab separated file) formats.

Time series data can be of multiple types based on the no of the columns or feature it consists of, i.e., it can be univariate, bivariate, or multivariate time series data.

Feature	Target
0.126883333	0.170658836128453
2.5651	2.55727688476776
3.820566667	3.81309328002033
0.162466667	0.181097542877393
0.2432	0.209858226866639
1.459316667	1.44116489682032
6.260033333	3.36416155338537
0.027766667	0.054410719958546
0.08395	0.0992957965599891
3.048916667	3.10140314614636
0.025033333	0.0998986102016724
0.020166667	0.0904578204051588
0.2351	0.197064865729522
0.2041	0.222716967810611
0.040933333	0.0765431465472989
0.1779	0.115356016343547
0.048666667	0.0243035042935704
0.0017	0.0594515595590129
0.0341	0.056383663068613
0.041966667	0.0656160089310237
0.010416667	0.0794416732805566
0.156333333	0.186960887203749
0.269333333	0.292587003431115
1.35465	1.2804489616346
0.038216667	0.0904704334552325
0.0467	0.0248679298084711

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292
893	3	Wilkes, Mrs. James (Ellen Needs)	female	47	1	0	363272	7
894	2	Myles, Mr. Thomas Francis	male	62	0	0	240276	9.6875
895	3	Wirz, Mr. Albert	male	27	0	0	315154	8.6625
896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22	1	1	3101298	12.2875
897	3	Svensson, Mr. Johan Cervin	male	14	0	0	7538	9.225
898	3	Connolly, Miss. Kate	female	30	0	0	330972	7.6292
899	2	Caldwell, Mr. Albert Francis	male	26	1	1	248738	29
900	3	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	female	18	0	0	2657	7.2292
901	3	Davies, Mr. John Samuel	male	21	2	0	A/4 48871	24.15
902	3	Ilieff, Mr. Ylio	male		0	0	349220	7.8958
903	1	Jones, Mr. Charles Cresson	male	46	0	0	694	26
904	1	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23	1	0	21228	82.2667
905	2	Howard, Mr. Benjamin	male	63	1	0	24065	26
906	1	Chaffee, Mrs. Herbert Fuller (Carrie Constance Toogood)	female	47	1	0	W.E.P. 5734	61.175
907	2	del Carlo, Mrs. Sebastiano (Argenia Genovesi)	female	24	1	0	SC/PARIS 2167	27.7208
908	2	Keane, Mr. Daniel	male	35	0	0	233734	12.35
909	3	Assaf, Mr. Gerios	male	21	0	0	2692	7.225
910	3	Ilmakangas, Miss. Ida Livija	female	27	1	0	STON/O2. 3101270	7.925
911	3	Assaf Khalil, Mrs. Mariana (Miriam")"	female	45	0	0	2696	7.225
912	1	Rothschild, Mr. Martin	male	55	1	0	PC 17603	59.4

Fig 1. Bivariate and Multivariate Time Series Data

- **Univariate Data:** Univariate data has an only single column; it can be either age or price etc. entries. E.g., Children's Height.

- **Bivariate Data:** Bivariate data consists of two columns where the first column will be of feature, and the other column is treated as the target value. Stock market data with time stamp is a mostly used example of bivariate data. Fig 1. Shows the pattern of bivariate data.

- **Multivariate Data:** In multivariate data, the data consists of multiple column values that serve as the feature space of the input parameter and single or multiple target values that serve as the output of the data at a particular time stamp. The feature space can be adjusted during the data pre-processing step, which will select only the features that are useful to achieve the resulting metric. This type of data is also represented as tabular format or csv, tsv format to store the records. Fig 1. Shows the example of Multivariate data type.

3.2.2 Machine Learning: Machine learning is the field of Artificial intelligence in which the system is being able to learn from the data through pattern recognization without programmed explicitly.

The primary objective of machine learning is to develop the program or algorithms that can extract the information from the data as that of a human being or better than a human being.

The process starts with the data about the particular problem that the program is going to solve. And then the data pre-processing is used to segregate the unwanted data which is out of the scope of that particular problem. After that, the learning phase of the algorithm starts, which learns from the data by observing the relationships, insights, and patterns in the data either by direct or indirect experience. The whole process is done without human assistance. Machine learning is divided into three categories as mentioned below:

- **Supervised learning:** In this category, the computer needs to learn the mapping from the input to output with given samples. This is commonly used as the maximum of the real world application has labeled data nowadays, e.g., Time series forecasting or regression problems, classification tasks with labeled data. As the target variables are given in this approach, the model has to find out the mapping between feature space and target variables. Then an evaluation is done based on comparing the actual values with the predicted ones.
- **Unsupervised learning:** This is entirely different from the supervised machine learning as in this the data is fed without the labeling, the algorithm all alone have to found the labels. These algorithms perform a different number of operations to find out the mapping to a number of operations to find out the mapping to in order to distinguish the data related to different classes, and this process is called clustering. Clustering is basically to find out the relationships between the data point either can be based on distance metric or by some other means.
- **Reinforcement learning:** This category uses the feature of both the categories mentioned above. This is based on a reward-based system with a goal-oriented mechanism. This uses the reward as a function to proceed further as a feedback variable. Nowadays, reinforcement learning plays a vital role in implementing

autonomous systems, e.g., autonomous vehicles, playing games, generating synthetic data, etc.

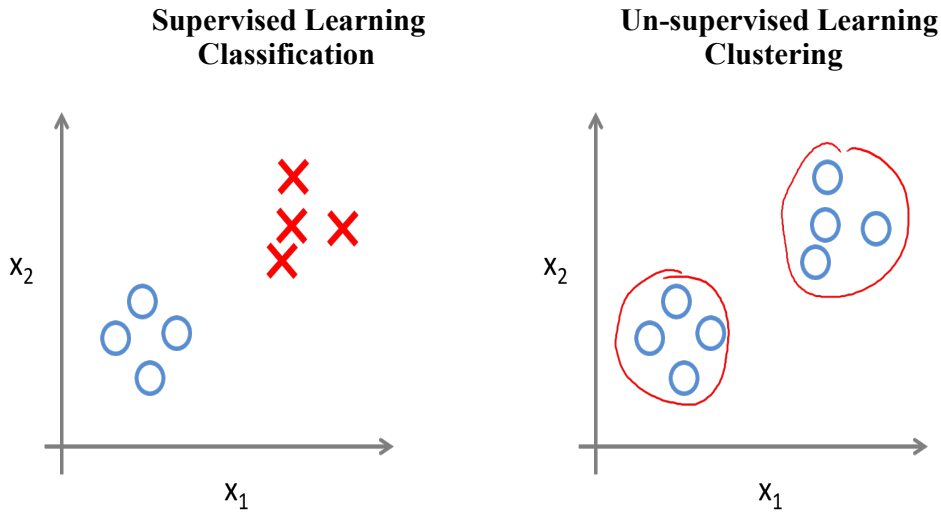

Fig 2: Supervised Learning with Labeled Data & Unsupervised Learning with Unlabeled Data

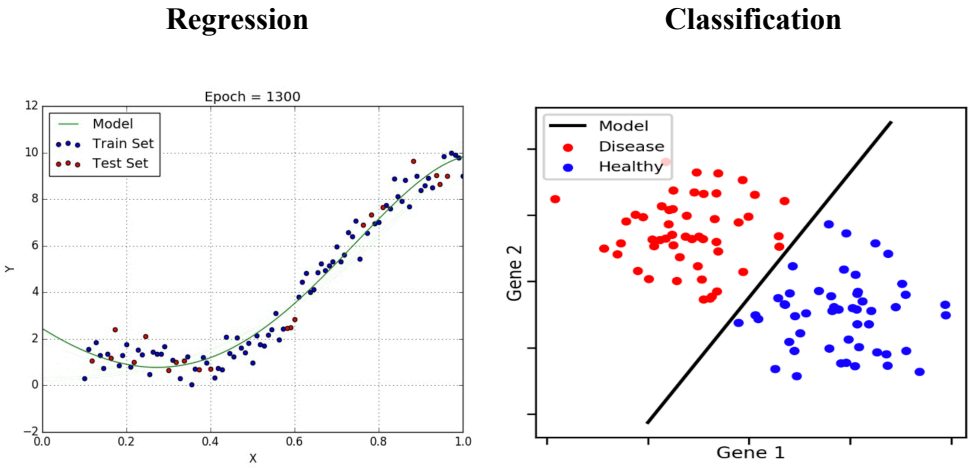

Fig 3: Example of Regression and Classification Task

Sometimes the Machine learning Algorithms are divided based on the output type of the system, which is nothing but the subcategories of the Supervised and Un-supervised Algorithms:

- **Regression:** This is a type of supervised learning. If the output is continuous rather than the discrete ones, then it is called a regression system. E.g., Time series, weather forecasting, stock market prediction, etc.

- **Classification:** This one also a type of supervised learning. In classification, the output variable is discrete, that signifies the classes in which the data/features lie. The problem to be tackled can be a binary classification or multi-classification.

- **Clustering:** This is the type of un-supervised learning mechanism. In this, the output is in the form of groups which are unknown before unlike the classification case. Predicting the Zero-day attacks or finding vulnerability/ anomalies in the system are its examples.

3.2.3 Deep Learning

Deep Learning is part of machine learning as it deals with getting more insights from the data to predict the result accurately and increase the overall performance. In Machine learning algorithms are not complex in structure as compare to the algorithms in Deep Learning. Just take an example a shallow (having less number of neurons and layers) neural network with backpropagation is consider under the machine learning whereas if we increase the neurons and layer mechanism it comes under the category of Deep Learning. The basic idea is to gain more information from the input data. Deep Learning nowadays can be easily found in all the real world tasks either related to natural language processing or computer vision or autonomous systems.

Deep learning is feasible due to the availability of the hardware resources to compute at a high level, such as GPUs, TPUs, Quantum Computers. Most commonly used machine learning models are Deep neural network, Deep Belief Networks, CNN, RNN, LSTM, GANs, Autoencoders.

Deep learning algorithms can also be used in all three categories, i.e., Supervised, Un-supervised, or Reinforcement Learning. However, the significant drawback is that initially, it requires a large amount of data as compared to machine learning algorithms. However, if there are large data available, then it outperforms the machine learning algorithms in many aspects. Because at a threshold point, the learning performance of the machine learning algorithms becomes constant, whereas in the case of deep learning it will increase concerning the data size.

3.3 Statistical Relational Learning

Whenever we talk about statistical learning, we try to visualize the data entries as points in a high dimensional space to understand its structure. The structural properties can also be retrieved from the equation associated with the data points. But this representation only tells us about structural properties, but there is still a lot of abstraction that hides its logical properties. These logical properties are crucial in solving complex problems. To deal with these hindrances, there comes the challenge to develop algorithms that have effective and robust reasoning about this statistical relationship in the data.

To deal with real-world problems, which has uncertainty at many levels, probabilistic models came into the place for effective learning and inference. These models made it possible to represent and learn based on the probabilistic semantics. These models are based on the graphical models to acquire the inference from the structure of the data. The SRL approach is mainly defined by either directed (Bayesian Network) or undirected graphical models, i.e., Markov Networks, also known as Markov Random Field. In our problem, we are using the Undirected Graphical approach to find out the relationship between data points to select the data point from the spatial environment.

The reason for choosing the undirected graphical model over the directed one is because when we represent the data entries in the form of graph nodes, we don't need the direction in order to propagate to find out the dependency or independence between the points. We use the Markov Random field to withdraw the insights from the data.

As shown in figure 4, the directed graph has a parent-child relationship, i.e., in figure 4 considering the node X_8, X_7 and X_3 serve as the parents, whereas the nodes X_{13}, X_9 are the children to node X_8 and node X_4, X_{12} are co-parents. Figure 5, is an example of Un-directed graphical Model Markov Random Field, where no directions are given, only the node and the nodes directly connected with it has a dependency.

If we use the directed graphical model, then the relationship changes whenever there is a shuffle in the input column, and that will also change the output. This is the reason to drop this methodology in order to have consistency over multiple units and shuffling of the input values.

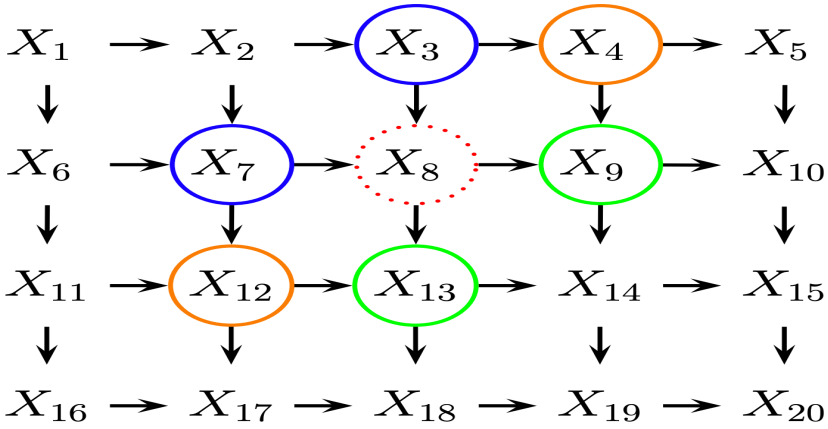

Fig. 4 Directed Graphical Model (Bayesian Network)

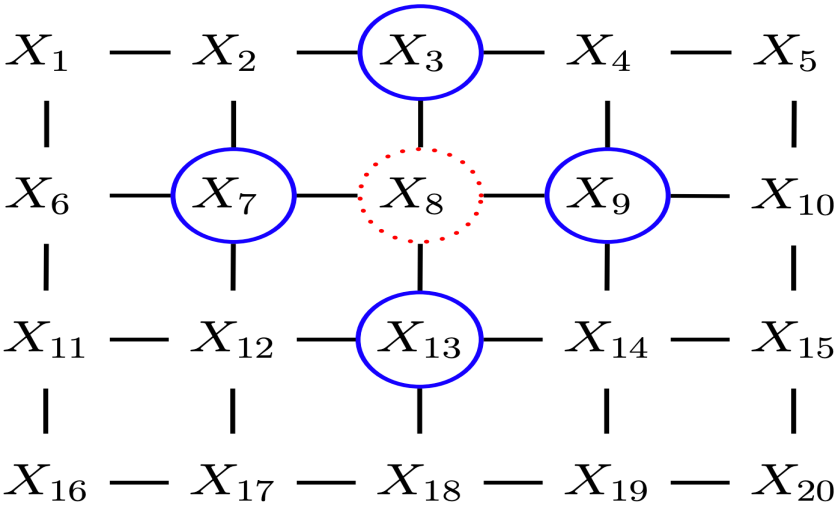

Fig. 5 Un-directed Graphical Model (Markov Random Field)

Markov Random Field: These are undirected graphical models which don't specify the edge orientations and have only its neighbor elements, in the undirected 2-D lattice, in Markov Blanket.

Mostly Markov Random Field follows the conditional independence property, i.e., no edge left between the two nodes in a graph if the third node gets eliminated then conditional independence holds. This is also called the Global Markov Property.

If there is no direct edge between two nodes, then we call it Pair-wise Markov Property. Determining the conditional independence is easier in Markov Random Field than the directed graphs.

3.4 Generative Modeling Method

A Generative Modeling framework has two components, a generator G, and a discriminator D.A generative modeling framework is has inherited features from the autoencoders. The Generator and Discriminator can be implemented using any deep learning algorithms conditioned that both components should use the same algorithm but just with reversed methodology from the first component, i.e., either Generator or Discriminator. The Generator uses the noise generated in terms of random variable z, over a uniform distribution as input, which serves as the latent space for the Generator. Based on the input it generates the data, xG, and then calculate the pData(xG), we try to maximize pdata(xG), which is the probability of the generated values over that data distribution.

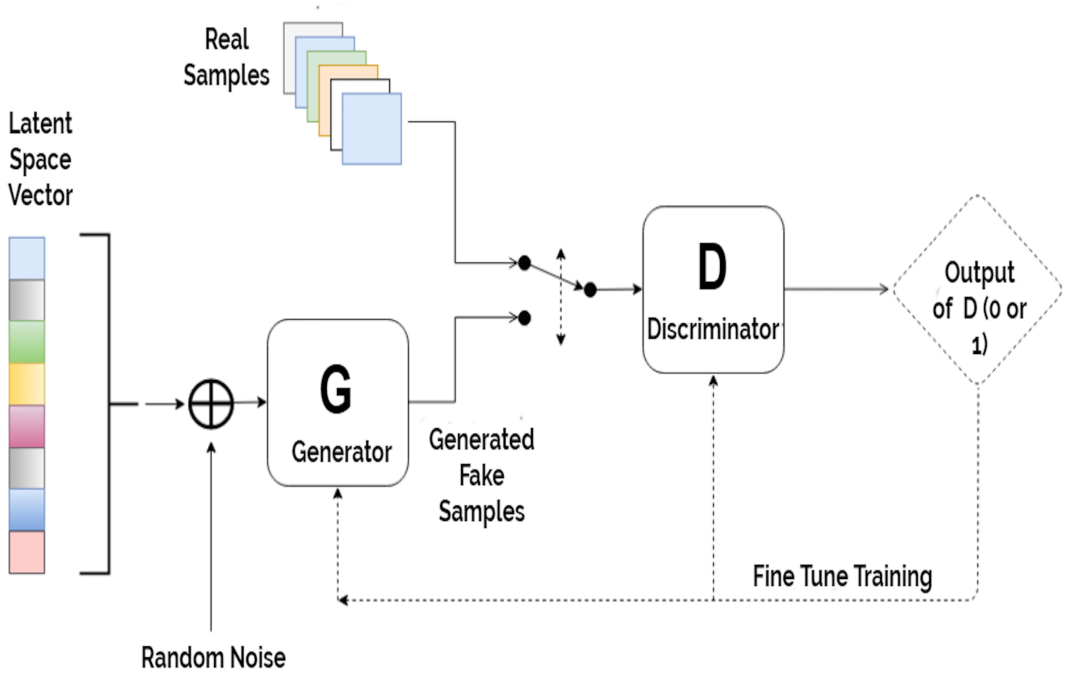

Fig. 6 Architecture of Generative Modeling

$$p_{data}(x_G) = \int_z p(x_G, z)dz$$
$$= \int_z p_{data}(x_G|z)p_z(z)dz$$

In order to perform the comparison or to have a relationship metric, Generator takes $p_z(z)$ as distribution for comparison.

The discriminator D simply takes data I as input and calculate pD(I) as output, which tells whether the given input is a part of real data or related to real data. The main function of the discriminator is to produce low probability output or minimize the classification loss when fed a “fake” (generated) image. Therefore, D is used to find out whether the data belongs to the real data or not.

Training of G and D is done adversially to improve by competing with each other. Sometimes they alternate each other role just in order to increase the performance of the system. By maximizing its score D(G(z)), The motive of the generator is ‘fool’ The discriminator that the outputs are from the true data distribution. The following optimization problem is achieved in the generator phase of training

$$\min_G V_G(D,G) = \min_G \left(\mathbb{E}_{z\sim p_z}[log(1 - D(G(z)))]\right)$$

Hence the combined loss of GAN can now be written as:

$$\min_G \max_D V(D,G) = \min_G \max_D \left(\mathbb{E}_{x\sim p_{data}}[logD(x)] + \mathbb{E}_{z\sim p_z}[log(1 - D(G(z)))]\right)$$

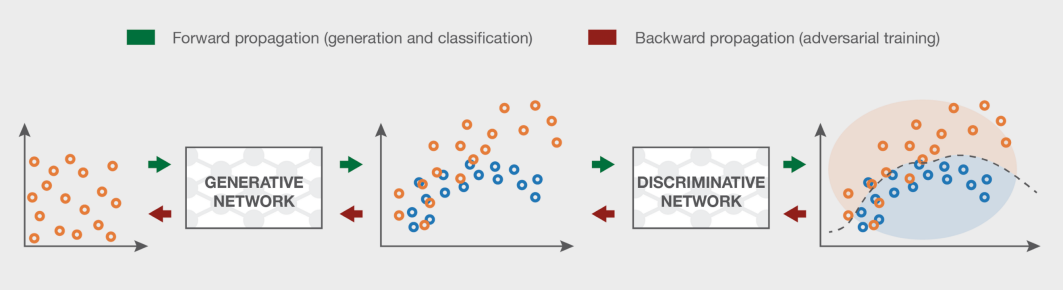

Fig. 7 Flow of data in Generative Modeling

First, the input of random variables is given to the Generator, which is also called noise as shown in figure 7.After that Generator is trained on these random variables in order to maximize the classification error, then the output from the Generator serves as the input to the discriminator along with the actual values. On which the discriminator performs the actions to minimize the classification error, which will serve as the basis metric for training.

3.5 FORECASTING MODELS

This section provides you the brief introduction of the ensemble model proposed in this paper and its components, i.e., SVM (Support Vector Machines), CNN 1d (Convolutional Neural Nets one dimensional), LSTM (Long Term Short memory).

A. SVM (Support Vector Machines)

A support vector machine (SVM) is a machine learning algorithm that is used for analyzing data for classification and regression analysis. SVM is a discriminative classifier algorithm that outputs an optimal hyperplane which categories a new example in the given labeled training data (supervised learning). It implements a learning algorithm, useful for recognizing patterns with the help of drawing the support vectors in complex data sets. The theory of SVM has originally been developed by Vapnik [11] and his co-workers at the AT&T Bell Laboratories, which was based on a separable bipartition problem. SVM is a type of learning system using a high dimensional feature space. It yields prediction functions that are expanded on a subset of support vectors. A version of an SVM for regression was proposed in 1997 by Vapnik et al. [12]. The Support Vector Regression (SVR) is based on the same principle as that of SVM for classification but with few changes. As the output is a real number, it becomes challenging to predict the information at hand, which has infinite possibilities. In SVM a margin of tolerance (epsilon) exists in approximation to the SVM. However, the primary objective is always the same: to minimize error, individualizing the hyperplane that maximizes the margin. The SVR equation is as follows:

$$y = wx + b,$$

$$\text{Solution} : \min(\|w\|)^2$$

$$\text{Constraints} : y_i w x_i b \leq \varepsilon, \qquad (1)$$

$$W x_i + b y_i \leq \varepsilon$$

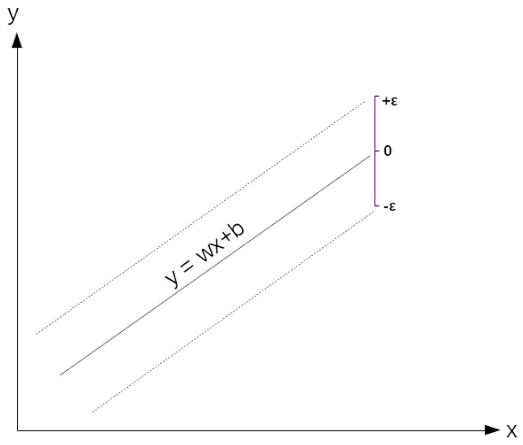

Fig. 8. Margin Estimation in SVM

Support vector regression is a special case of SVM's large margin kernel methods which are used for classification to regression analysis. The most important parameter in SVR is the kernel parameter. After scaling the data set, the next step is to use a kernel method or function for creating the model. A kernel is a similarity function. It is a function that is provided to a machine learning algorithm. It takes two inputs and returns out how similar they are. Among the various kernels available, the one chosen for our model is the linear kernel. Linear kernel having fewer parameters as compared to the most used RBF kernel makes it easier to train besides gives us good accuracy of 99.81% in the dataset of the single home system. Another essential factor to take under consideration is that SVM with the linear kernel is less prone to overfitting as compared to non- linear kernels in cases where the dataset is big as in our case.

Another three parameters include the Regularization (which tells the SVR optimization how much we want to avoid misclassifying each training example), Gamma (which defines how fat the influence of single training example reaches), margin (which is a separation of the line to the closest class points). The most important advantage of SVR is that optimality is guaranteed. As in Neural Network multiple solutions associated with local minima as a result of this, it becomes less robust over different samples. Semi-Supervised learning can be implemented by SVM. For this reason, SVMs are regarded as a useful tool for effectively complementing information gained from classical linear regression techniques.

B. Convolutional neural networks

There is a keen growing interest in neural-based-learning because of its accurate and superior performance in many applications, such as speech and image/video analysis. The recent hype of deep neural networks (DNN) is because of the availability of a large amount of supervised learning based labeled data and more efficient hardware nowadays. The most common architectures of neural networks are Convolutional Neural Network (CNN) [13] and Recurrent Neural Network (RNN). CNN is used to get the spatial insight from pixels in an image in a 2-D or 3-D space. RNN is used to identify patterns in a sequential manner from audio/speech or time series data which can be represented in 1-D. Both CNN and RNN are multilayer neural networks. For our work here, we have used the Convolutional Neural Network so that we will focus on that in detail.

A convolutional neural network (CNN) is a type of artificial neural network that uses perceptron, a machine learning unit algorithm, for supervised learning, to analyze data. CNN has shown extraordinary performance in visual recognition tasks like recognizing traffic signs, faces, and hand-written digits. There has been a large number of recent works which help us to understand CNN better. Therefore, CNN is a unique form of the feedforward neural network (FNN), also known as the multi-layer perceptron (MLP) with backpropagation. It was proved in [14] that FNNs are capable of universal approximation, which serves as a building block of the CNN.

For our research work here, we use CNN 1D model in which input matrix is of dimension [W_1,1] since we want the model to predict the load forecasting based on a dataset containing weather and other parameters. The layers used in ConvNets are as follows:

1) Input Layer: The input layer takes the matrix of size [W_1,1].

2) Convolution Layer: This layer is used to compute the output from neurons which are connected to local regions, computing a dot product between the weights of the input neurons and the small region in which they are connected to the input volume. The working of Conv layer is as follows:

1). Accepts a volume of size [W_1, 1].

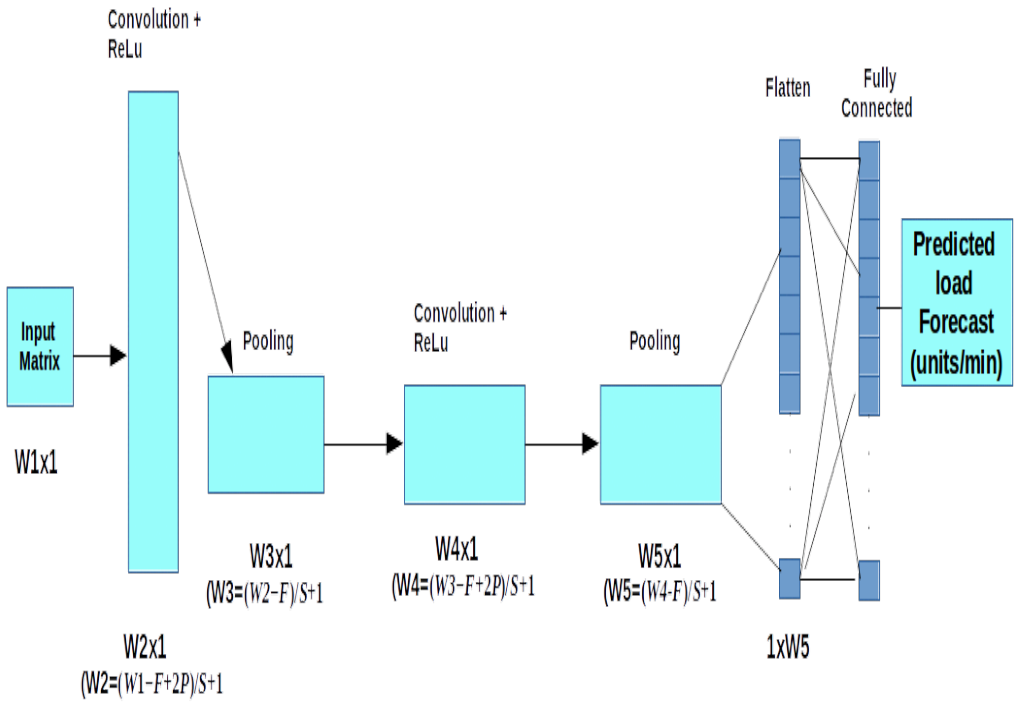

Fig. 9. Architecture of CNN

2). Set the values of 4 hyper-parameters that are:

a) Number of filters (K)

b) their spatial extent (F)

c) the stride (S): value with which we stride the filter.

d) zero padding (P): to pad the input values with zeroes around the border or not.

3) Produces a volume of size $[W_2,1]$ where:

$$W_2 = (W_1F+2P)/S+1 \quad (2)$$

4) In the output volume, the d_{th} depth slice (of size $[W_2,1]$) is the result of performing a valid convolution of the d_{th} filter over the input volume with a stride of S and then offset by d_{th} bias.

Then we apply an element-wise activation function. The activation function is used to determine the output of the neural network. The activation function is basically divided into two parts: Linear Activation function and Non-Linear Activation Function. For our purpose here, we will use ReLU activation function, which is used to add non-linearity for our data [12].

$$R(Z) = \max(0, Z) \quad (3)$$

ReLu is basically half rectified. Its value is zero when Z is less than zero and equal to Z when Z is above or equal zero. Its range is zero to infinity.

3) Pooling Layer: Generally we insert a pooling layer in between consecutive Conv layers in a ConvNet architecture. The function of this layer is to reduce the spatial size of the feature map to reduce the number of parameters, and hence to control overfitting. The pooling layer operates on every depth slice of input and resizes it spatially, using MAX operation. The pooling layer :

1). Accepts a volume of size [W_2,1] from the above layer.

2). Set the values of 2 hyper-parameters that are:

a) the spatial extent (F)

b) the stride (S): value with which we stride the filter.

3) Produces a volume of size [W_3,1] where:

$$W_3=(W_2F+2P)/S+1 \quad (4)$$

4) Introduces zero padding since it computes a fixed function of the input.

5) Note that it is not usual to use zero-padding for Pooling layers.

4) Flattening Layer: The basic function of this layer is convert the matrix obtained in the form of [W_3,1] to [1, W_3] so that it can be passed to the input of neural network in the fully connected layer.

5) Fully-Connected Layer: Neurons of a fully connected layer are connected to all activations in previous layers, as seen in a regular neural network. The matrix after the flattening step is passed into this layer connecting a network similar to Artificial Neural Network with the property that it is fully connected in the hidden layers. The output of the final fully connected layer is mapped into one neuron which learns by back-propagation which of the final fully connected neuron.

C. Recurrent Neural Networks

Recurrent Neural Networks (RNN) are a powerful and robust type of neural network with internal memory. In RNN the information cycles, forming loops. When it takes a decision, it takes into consideration the current input and also what it has learned from input previously it received. All RNN networks have the form of chain repeating models of neural networks with a straightforward structure such as a single tanh layer. RNN but usually have short term memory. An extension to RNN is Long-Short Term

Memory(LSTM) network. LSTM enable RNN to remember their inputs over a long period of time. LSTM contain their information in memory, which is just like computer memory. In LSTM there are three gates: input (to decide whether or not to let new input in), forget (delete the information that's not important), output (decide whether to make it impact the current time step). These gates are sigmoids, that is the range from 0 to 1. This enables them to do backpropagation with them. The working is as per the algorithm 1. is given below.

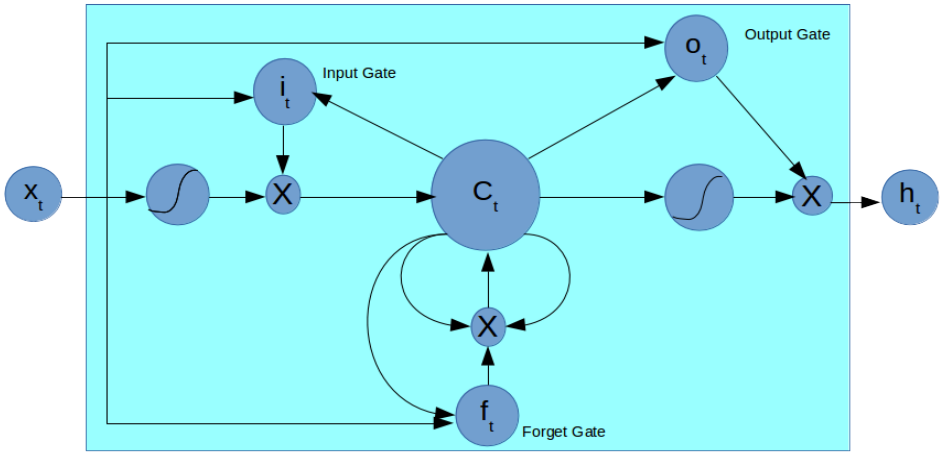

Fig. 10. Architecture of LSTM's single node

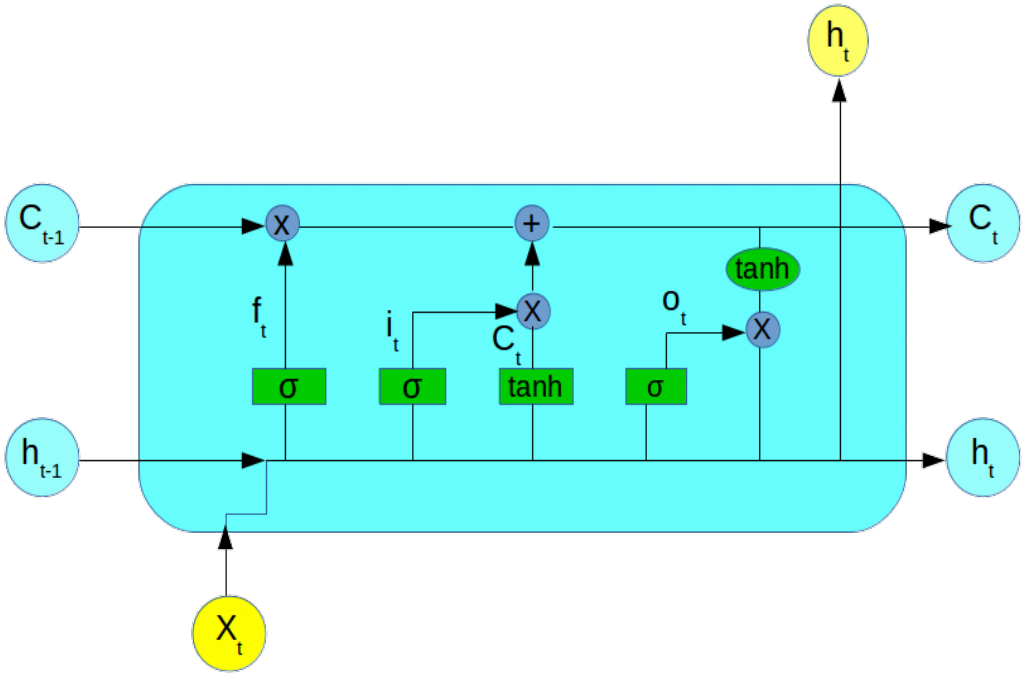

Fig 11. LSTM Cell State Working

Algorithm 1 LSTM Cell State Working

Algorithm 1 LSTM Cell State Working

Input: $\mathbf{X_t}$: Input Array, $\mathbf{C_{t-1}}$: Memory Unit,
$\mathbf{h_{t-1}}$: Previous cell output
Output: $\mathbf{C_t}$: Updated Memory unit, $\mathbf{h_t}$: Cell output

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + bf) \tag{5}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{6}$$

$$C'_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{7}$$

$$C_t = f_t.C_{t-1} + i_t.C'_t \tag{8}$$

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o \tag{9}$$

$$h_t = O_t.tanh(C_t) \tag{10}$$

procedure PREDICTION(An array of dimension [48,1])
    if f_t==true **then** *Update the memory unit*

$$a_t = C_{t-1} \otimes f_t \tag{11}$$

    else
        forget it $\rightarrow$ a_t=C_{t-1}
    end if
Calculate i_t & C'$_t$

$$m_t = i_t \otimes C'_t \tag{12}$$

$$C_t = m_t \oplus a_t \tag{13}$$

Calculate O_t & h_t
end procedure

CHAPTER 4

EXPERIMENTAL SETUP

In this chapter, an overview of the experimentation process is given as — first, the introduction of the dataset being used. Then, the implementation is discussed.

This thesis focuses on mitigating the difficulties discussed in introduction part by means of applying the statistical models on the benchmark dataset that also take into account the weather condition as well as the individual component energy generation power either by solar power or by biogas systems with a sampling rate of 1 minute.

The proposed approach in terms of a statistical model which is based on machine learning and deep learning. It ensembles the results of the machine learning model (SVM) and deep learning models (CNN 1D and LSTM) with a shallow neural network. The previous work for load forecasting with neural network ensembling works good [4], but there are some drawbacks of using them as they are prone to overfitting that leads to the wrong prediction over the different location/components. They also take more time to predict the results because of more number of parameters, especially in case of short or middle-term load forecasting. In our system, we use SVM for forecasting the demand for the short-term and middle-term load forecasting by limiting the size of weights for deep learning models to 0 and by adding constraints in the model. It leads to more accuracy and less prone to overfitting due to less number of parameters and regularization and also requires less training time. For long-term load forecasting, we use the ensemble model SVM and CNN 1D and LSTM. Here we use deep learning models with machine learning model because as the data increases the learning curve of the machine learning models becomes constant, but for deep learning models, it increases with increase in data. Model is being tested on the same data with same split ratio, and it outperforms the existing models, or the individual model tested on the same dataset (1.5% for 65000 extra readings) in accuracy whereas the CNN-1D is used to mitigate the overfitting in the system.

A. Dataset

The proposed methodology was implemented a benchmark dataset named UMass Smart* Dataset - 2017 release [21]. The dataset contains electric usage for the period 2014-2016. We use house data sets of 2016 with time stamp 1 minute with weather conditions too. We use the dataset with 1079542 readings. Attributes that the data contains are a timestamp(1 min), Use(kW), Gen(kW), Appliances usage(kW), Solar

power generation(kW),icon humidity, visibility, summary, apparent- temp, pressure, windspeed, cloudCover, windbearing, precip- Intensity, dewpoint, PrecipProbability. The dataset for training and testing was split into 70% training, 15% testing, 15% validation data.

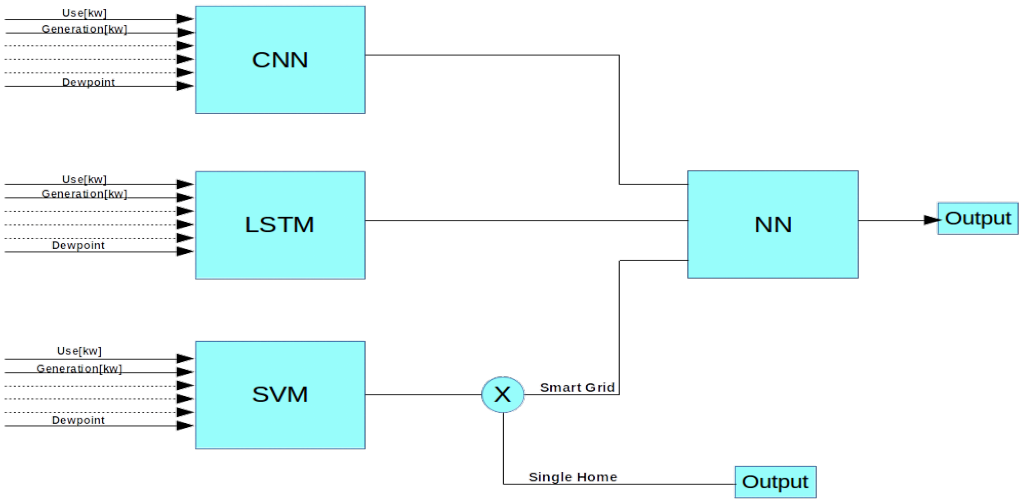

Fig. 12 Architecture of the Proposed Model for Forecasting

B. Implementation Details

- **Hardware Specification**

a) CPU: Intel Core i9- 9900K (8 Cores, 16 Threads)

b) RAM: 64 GB

c) GPU: 2* NVIDIA GEFORCE RTX 2080Ti (11 +11 GB VRAM)

d) Storage: 512 GB PCIe NVMe SSD, 2 TB HDD.

1) For Synthetic Data generation

In order to generate the synthetic data, noise is fed to the generator along with the relational insights from the data obtained from the SRL model is trained on the real data. The generator generates an output that will serve as the input to the discriminator along with real-world data and if the classification error of the real world data and the synthetic data is less than the discriminator except for the values otherwise reject them. The SRL model helps the generator to get a push start by feeding the input with relational inferences. The input is fed in the form of the 2D lattice. We can also configure the latent space that we are providing the SRL and generator to create the synthetic data just by increasing the number of historical inputs. This will be the input memory in the form of latent space for the generator to extract the insights in a better manner based on the historical data values.

2) **For Forecasting the Time Series Data**

Preprocessing Phase: Data were preprocessed before feeding to the model. Preprocessing step contains removing the NaN entries, formatting the timestamp, factorizing the data, reshaping the data concerning the model (adding extra dimensions to the data).

Training Phase: After the preprocessing phase and validation of constraints, the input is fed into the model, and all three internal models will train with different input dimensions based on their algorithms. Data fed into the model and all the three internal model takes the same data as input, but the shape of the data varies in all the models. After training of the model, the result predicted by all the models is fed into the shallow neural network. And then the final result will be given by the neural network output layer. Internal processing of models are as follows:

- **SVM:** In support vector machine input is in the form of a tuple with all the column values except the target value. After that parameter are calculated to find out the curve/margin that fits the data. This curve/margin line then further used to predict the target variable as per the input is given.

- **CNN 1D:** In Convolution 1D, the input is in the form of an array of dimension [48,1].This array is fed into convolution layer with hyper-parameter values stride=2, pooling=2 and activation function or non-linear transformation using ReLU activation function i.e. linear rectifier unit. After the convolution layer, the results are passed to the pooling layer where the size of the input will be decreased because this layer will work as feature selection by means of filtering the data with max pooling. The output provided by this pooling layer will be sent to the fully connected layer, which is nothing but the hidden layer containing 32 nodes that are used to fabricate the output.

- **LSTM:** In LSTM the input vector of the form [48,1] is passed, which is first concatenated with the previous output of the LSTM unit. Then we perform pointwise operations (xor and or operations) by making through the activation functions (sigmoid and tanh), and the final output is generated by the concatenating the outputs from all the gates. This output is passed to the next LSTM unit. Along with this output, the memory unit also gets updated by performing a pointwise operation after applying the activation function.

CHATPER 5

RESULT AND DISCUSSION

5.1 Evaluation Metrics

We worked on different regression models in order to find out which model works well on our dataset. The model is selected based on the following parameters:

Mean Absolute Error (MAE): In machine learning, mean absolute error (MAE) is a model evaluation metric often used with regression models. The mean absolute error is formulated as an average of the difference between the predicted value and true value.

$$\text{Absolute Error} = \|e_i\| = \|y_i - x_i\|$$

$$\text{Mean Absolute Error} = \sum_{i=1}^{n} \frac{|e_i|}{n} \quad (14)$$

where y_i is the prediction and x_i is the true value.

Accuracy: Accuracy is just a metric by which you examine how good is your machine learning model. In statistics it is formulated as:

Accuracy = Number of correct predictions/Total Number of predictions

Coefficient of Correlation (r): The strength of association between two variables is called a Coefficient of Correlation. The most common type is the Pearson product-moment correlation coefficient, is used to measure the linear correlation between variables X and Y. Formula for computing The Pearson correlation coefficient is as follows:

$$r = \frac{\sum xy}{\sqrt{\sum x^2 \sum y^2}} \quad (15)$$

Where $x=x_i - x$, x_i is the x value for i^{th} observation , x is the mean x value, $y=y_i - y$, y_i is the y value for i^{th} observation , and y is the mean y value.

Coefficient of Determination (R): The coefficient of determination is a key output of regression analysis. It is the proportion of variance in dependent variable predicted by the independent variable and is denoted by r2.

AUC-ROC Curve (AUROC): AUC stands for Area Under the Curve whereas ROC stands for Receiver Operating Characteristics. AUC part is used to find out the degree of separability and ROC is just a probability curve.

5.2 Results

5.2.1 Synthetic Data Generation Results

The AUROC values obtained from the synthetic data and real data by varing the training data size and increasing the length of the time series data, i.e., number of features. We find out that the difference is quite less and the synthetic data obtained from the whole process is acceptable. After that we visualize the synthetic data on the day and night basis.

Training Set Size	Data Type	Length of Time Series Data				
		10	20	30	40	42
1000	Synthetic	0.966±0.013	0.971±0.21	0.979±0.54	0.987±0.43	0.988±0.49
5000	Synthetic	0.968±0.12	0.973±0.19	0.981±0.072	0.991±0.38	0.990±0.14
10000	Synthetic	0.939±0.09	0.970±0.38	0.980±0.5	0.986±0.42	0.992±0.79
20000	Synthetic	0.981±0.034	0.976±0.25	1.0±0.0 0	0.989±0.27	0.995±0.38
50000	Synthetic	0.978±0.096	0.983±0.91	0.989±0.75	0.988±0.19	0.998±0.94
1000	Real	0.988±0.055	0.986±0.07	1.0±0.0 0	1.0±0.0 0	1.0±0.0 0
5000	Real	0.981±0.29	0.983±0.34	0.992±0.26	0.994±0.35	1.0±0.0 0
10000	Real	0.996±0.31	0.989±0.05	1.0±0.0 0	1.0±0.0 0	0.999±0.06
20000	Real	0.983±0.357	0.988±0.82	0.989±0.73	1.0±0.0 0	1.0±0.0 0
50000	Real	1.0±0.0 0	0.992±0.95	0.991±0.86	1.0±0.0 0	1.0±0.0 0

Table 1. AUROC value for synthetic and real data with different test size and time series length

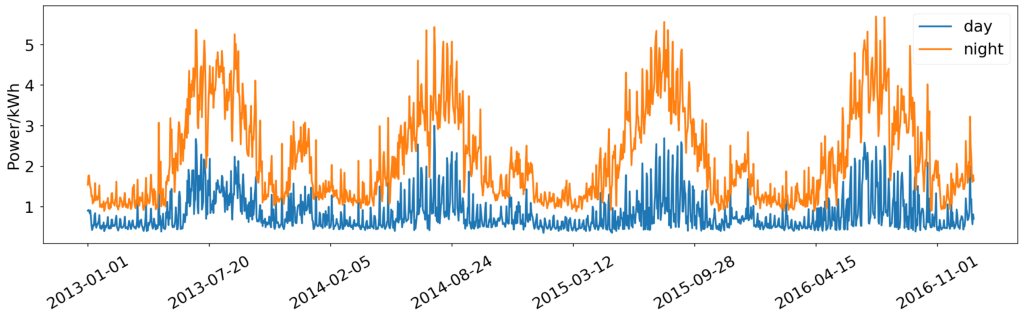

Fig. 13 Real Data Visualization

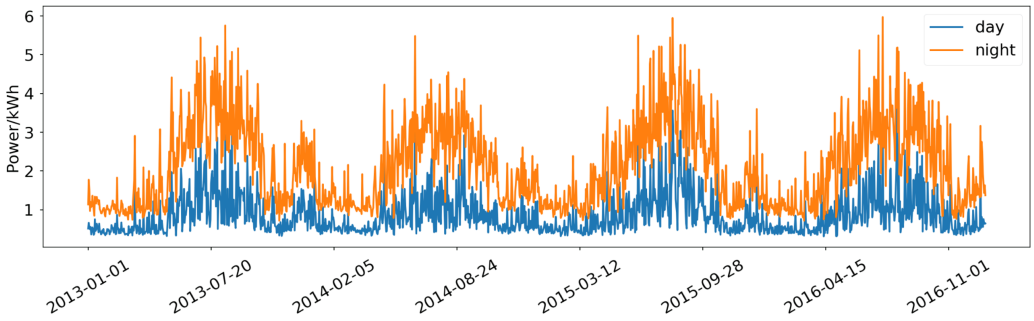

Fig. 14 Synthetic Data Visualization

Loss visualization as shown in figure 15 also shows that the generator function loss is decreasing and discriminator function loss is increasing to minimise the classification error of the output of generator and output of discriminator.

Fig. 15 Loss visualization of Generator and Discriminator

5.2.2 Forecasting Model's Results

The major advantage of SVM is optimality is guaranteed since the optimality problem is convex. This is an advantage over the neural networks as it has multiple solutions associated with local minima therefore it may not be robust over different samples. Another advantage of SVM is that it is useful for both linearly separable (hard margin) and non-linearly separable (soft margin). This makes it robust over linear regression.

Model Name	r	R	Accuracy	MAE
Linear model	0.98	0.96	96.4	0.001
Logistic Reg	0.99	0.98	97.1	0.0007
Random Forest	0.48	0.23	56.3	15.6
SVM	0.99	0.98	99.3	0.0005
ANN	1	1	99.2	0.0006

Table 2. Results obtained from the Machine Learning Algorithms

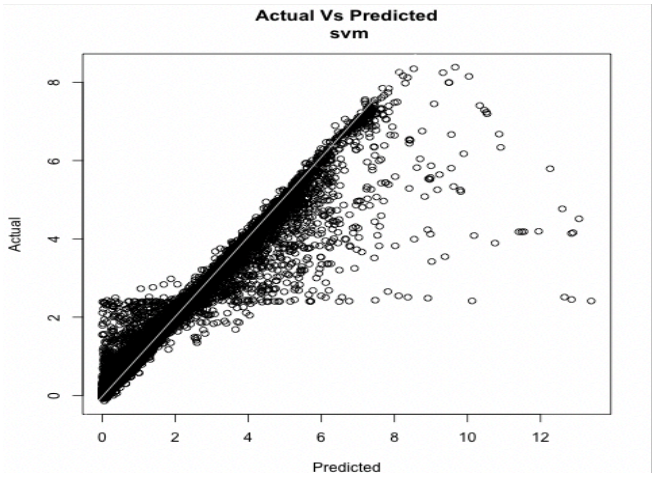

Fig. 16 SVM Prediction

With increase in data the effect on accuracy on all the above models is negligible (no change at all after the dataset with total no of rows greater than 200000). This is due to the degradation (constant) of performance of older machine learning algorithm to deal with this issue we use the deep learning algorithms that completely overcome this issue and give us significant increase in accuracy as shown in the table:

Model Name	Training Dataset(No. of Rows)	MAE	Accuracy
LSTM	200000	0.018	32.3%
LSTM	270000	0.007	33.63%
LSTM	350000	0.0016	36.12%
LSTM	420000	0.0013	37.06%
LSTM	500000	0.0011	37.89%
LSTM	570000	0.0010	38.47%
LSTM	650000	0.0009	39.38%
LSTM	720000	0.0009	39.97%
LSTM	800000	0.00089	40.65%
SVM	200000	0.0005	99.3%
SVM	270000	0.00054	99.3%
SVM	350000	0.00054	99.27%
SVM	420000	0.00056	99.21%
SVM	500000	0.00057	99.16%
SVM	570000	0.00057	99.08%
SVM	650000	0.00057	98.73%

Table 3. Performance of LSTM and SVM with varying the data Size

When we visualize the result from Table 3, we found out the basic property of Deep Learning Algorithm satisfies, i.e., performance of ML Algorithms degrades/remain constant with increasing the data size whereas for DL algorithms it increases as shown in figure 17.

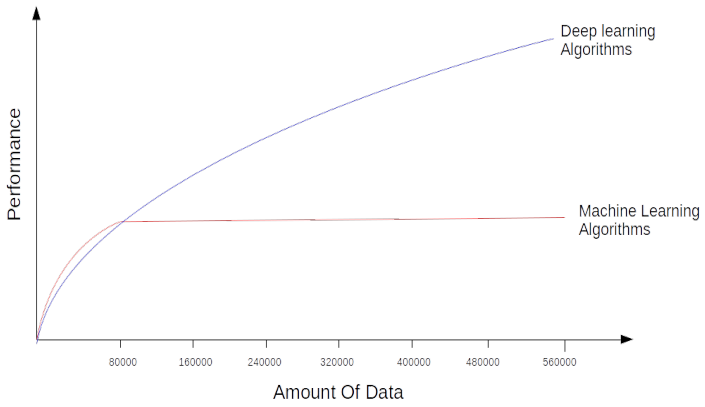

Fig. 17 Performance curve for ML and DL Algorithms

Whereas the performance of CNN-1D is not upto mark so we use that one as a model in our ensemble model to deal with the overfitting issue.

Model Name	**MAE**	**Accuracy**
CNN-1D	3.587	5.13%

Table 4. Performance of CNN-1D

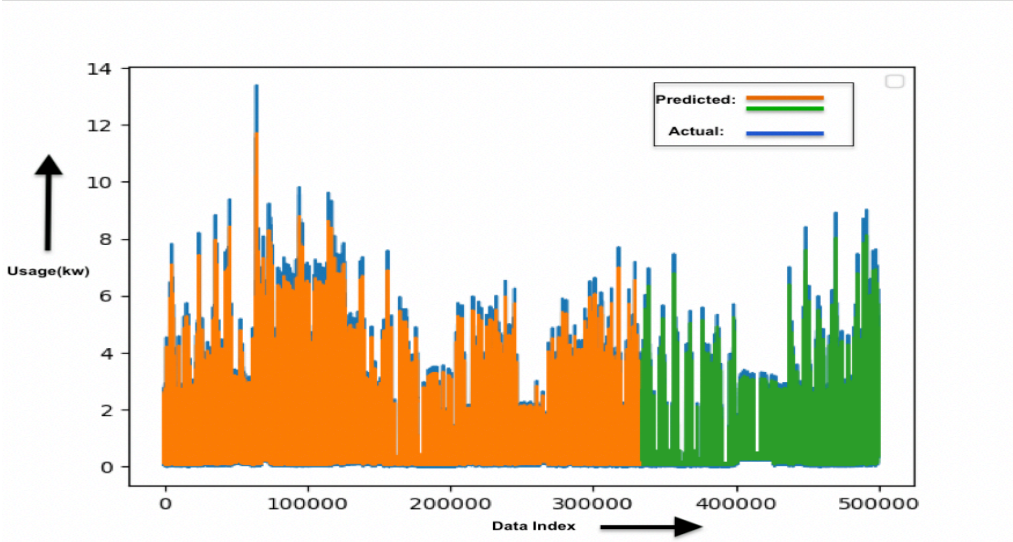

Fig. 18 LSTM Prediction.

The overall results from the model are as shown in figure 19 and in table:

Model Name	r	R	MAE	Accuracy
Ensemble Model	0.98	0.96	0.00073	95.6%

Table 5. Overall System Performance

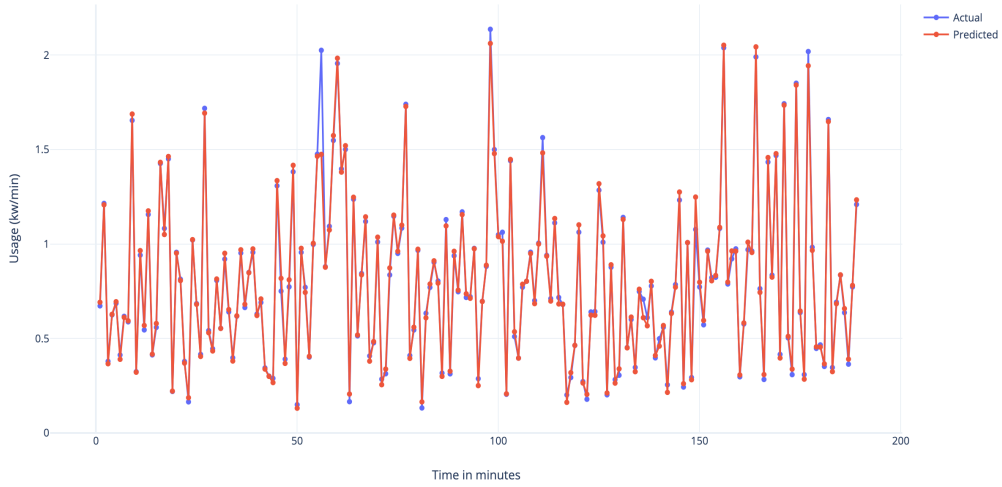

Fig. 19 Overall System Prediction

CHAPTER 6
CONCLUSION

6.1 Key Findings

The Key contribution of this thesis is to implement a solution that can work on both the short-term and long-term load forecasting and gives us significantly better accuracy without overfitting as compared to an existing solution. Also, in the case of small data, it can generate synthetic data in order to maintain the performance of the whole and for better forecasting. The presented solution with two-tier architecture is to calculate the demand for smart grid over its components/building as well as an aggregate level. To implement a better solution all the experiments are being carried out on a benchmark data with weather data in order to get better results as weather data is an important factor when we consider the energy consumption with different models. And after comparison of the models being trained, we found out that for short-term load forecasting SVM performs better with fewer drawbacks including the one of not being good with the larger amount of data. We ensemble SVM with the LSTM and CNN 1D by feeding their result to a shallow neural network to get the final result to make an adaptive model with respect to the size of the data that is less prone to overfitting. This solution will give us a significant improvement over the existing solutions with the scope not only limited to short or long-term load forecasting.

6.2 Contribution

The research contribution of this thesis is as follows:

• In order to deal with low availability of data due to privacy or other concerns, we proposed a model to generate the synthetic data.

• A single ensemble model that can predict the demand for both shot-term and long-term load forecasting.

• The proposed scheme is also designed to deal with adaptiveness with respect to the size of data mean- while maintaining the accuracy and performance.

• By considering the weather condition features too for training the models makes the model more robust as compare to existing models because weather condition plays a key role in demand prediction [21,22].

6.3 Future Work

- To create the more optimised and efficient Data generation model that can generate data from small data set, i.e., Dataset size<500 rows.
- Improve the Generator part in order to deal with data other than time series, i.e., classification data, relational data or unstructured data, e.g. JSON, XML formats.
- Generalizing the forecasting model such that it can also be used in the other regression problem in same manner as in smart grid.
- Optimising the output of the forecasting model with respect to the rate change of the environment variables that affects it with differential equation.
- Training of the proposed synthetic data generator on more datasets related to time series.

REFERENCES

1. Mocanu, Elena, et al. "Deep learning for estimating building energy consumption." Sustainable Energy, Grids and Networks 6 (2016): 91-99.

2. Amarasinghe, Kasun, Daniel L. Marino, and Milos Manic. "Deep neural networks for energy load forecasting." 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE). IEEE, 2017.

3. Raza, Muhammad Qamar, and Zuhairi Baharudin. "A review on short term load forecasting using hybrid neural network techniques." 2012 IEEE International Conference on Power and Energy (PECon). IEEE, 2012.

4. Kuo, Ping-Huan, and Chiou-Jye Huang. "A high precision artificial neural networks model for short-term energy load forecasting." Energies 11.1 (2018): 213.

5. Seetha, Hari, and R. Saravanan. "Short term electric load prediction using fuzzy BP." Journal of computing and information technology 15.3 (2007): 267-282.

6. Ahmad, A. S., et al. "A review on applications of ANN and SVM for building electrical energy consumption forecasting." Renewable and Sustainable Energy Reviews 33 (2014): 102-109.

7. Ho, Kun-Long, Y-Y. Hsu, and C-C. Yang. "Short term load forecasting using a multilayer neural network with an adaptive learning algorithm." IEEE Transactions on Power Systems 7.1 (1992): 141-149.

8. Bakirtzis, A. G., et al. "A neural network short term load forecasting model for the Greek power system." IEEE Transactions on power systems11.2 (1996): 858-863.

9. Lijesen, D. P., and J. Rosing. "Adaptive forecasting of hourly loads based on load measurements and weather information." IEEE Transactions on Power Apparatus and Systems 4 (1971): 1757-1767.

10. Taylor, James W., and Roberto Buizza. "Neural network load forecasting with weather ensemble predictions." IEEE Transactions on Power systems17.3 (2002): 626-632.

11. Sapankevych, Nicholas I., and Ravi Sankar. "Time series prediction using support vector machines: a survey." IEEE Computational Intelligence Magazine 4.2 (2009): 24-38.

12. Vapnik, Vladimir, Steven E. Golowich, and Alex J. Smola. "Support vector method for function approximation, regression estimation and signal processing." Advances in neural information processing systems. 1997.

13. Bengio, Y., I. J. Goodfellow, and A. Courville. "Deep learning'An MIT Press book in preparation." Draft chapters available at (2015).

14. LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." The handbook of brain theory and neural networks3361.10 (1995): 1995.

15. Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

16. Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).

17. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

18. Hinton, Geoffrey E., et al. "Improving neural networks by preventing co-adaptation of feature detectors." arXiv preprint arXiv:1207.0580 (2012).

19. Lauret, Philippe, et al. "Bayesian neural network approach to short time load forecasting." Energy conversion and management 49.5 (2008): 1156-1166.

20. Khan, Ahsan Raza, et al. "Load forecasting, dynamic pricing and DSM in smart grid: A review." Renewable and Sustainable Energy Reviews 54 (2016): 1311-1322.

21. UMass Smart* Dataset -2017 release,
URL: http://traces.cs.umass.edu/index.php/Smart/Smart

22. Vu, Dao H., et al. "Intra-hour and hourly demand forecasting using selective order autoregressive model." 2016 IEEE International Conference on Power System Technology (POWERCON). IEEE, 2016.

23. Arumugam, U., N. M. Nor, and M. F. Abdullah. "A review of triplen harmonics estimation and forecasting techniques applied to GDC-UTP distribution network." 2011 National Postgraduate Conference. IEEE, 2011.

24. Parras-Gutierrez, Elisabet, et al. "Short, medium and long-term forecasting of time series using the L-Co-R algorithm." Neurocomputing 128 (2014): 433-446.

25. Akdemir, Bayram, and Nurettin Çetinkaya. "Long-term load forecasting based on adaptive neural fuzzy inference system using real energy data." Energy Procedia 14 (2012): 794-799.

26. Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

27. Anderson, Jason W., et al. "Synthetic data generation for the internet of things." 2014 IEEE International Conference on Big Data (Big Data). IEEE, 2014.

28. Hu, Joseph W., et al. "Distribution-Driven, Embedded Synthetic Data Generation System and Tool for RDBMS." 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW). IEEE, 2019.

29. Myung, Rohyoung, et al. "Elaborate Synthetic Data Generation for Internet of Things Services at Smart Home Environment." 2016 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2016.

30. Patki, Neha, Roy Wedge, and Kalyan Veeramachaneni. "The synthetic data vault." 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE, 2016.

31. Santos, Miriam Seoane, et al. "Generating Synthetic Missing Data: A Review by Missing Mechanism." IEEE Access 7 (2019): 11651-11667.

32. Esteban, Cristóbal, Stephanie L. Hyland, and Gunnar Rätsch. "Real-valued (medical) time series generation with recurrent conditional gans." arXiv preprint arXiv:1706.02633 (2017).

33. Koller, Daphne, et al. Introduction to statistical relational learning. MIT press, 2007.

34. Bishop, Christopher M. Pattern recognition and machine learning. springer, 2006.

Paper Communicated

"Two-Tier Ensemble Model for Demand Prediction in Smart Grid Environment" in IEEE Global Communications Conference (GLOBECOM) 2019.

Time Series Forecasting using SRL based Deep Learning in Smart Grid

ORIGINALITY REPORT

8%	5%	6%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijser.in Internet Source	<1%
2	Saman Sarraf, Ghassem Tofighi. "Deep learning-based pipeline to recognize Alzheimer's disease using fMRI data", 2016 Future Technologies Conference (FTC), 2016 Publication	<1%
3	stattrek.com Internet Source	<1%
4	A.S. Ahmad, M.Y. Hassan, M.P. Abdullah, H.A. Rahman, F. Hussin, H. Abdullah, R. Saidur. "A review on applications of ANN and SVM for building electrical energy consumption forecasting", Renewable and Sustainable Energy Reviews, 2014 Publication	<1%
5	Vahid Moosavi, Ali Talebi, Mohammad Hossein Mokhtari, Seyed Rashid Fallah Shamsi, Yaghoub Niazi. "A wavelet-artificial intelligence	<1%

	fusion approach (WAIFA) for blending Landsat and MODIS surface temperature", Remote Sensing of Environment, 2015 Publication	
6	www.ijfcc.org Internet Source	<1%
7	www.irjet.net Internet Source	<1%
8	ieeexplore.ieee.org Internet Source	<1%
9	"Cognitive Informatics and Soft Computing", Springer Science and Business Media LLC, 2019 Publication	<1%
10	D. Lijesen, J. Rosing. "Adaptive Forecasting of Hourly Loads Based on Load Measurements and Weather Information", IEEE Transactions on Power Apparatus and Systems, 1971 Publication	<1%
11	hrcak.srce.hr Internet Source	<1%
12	M. Q. Raza, Z. Baharudin. "A review on short term load forecasting using hybrid neural network techniques", 2012 IEEE International Conference on Power and Energy (PECon), 2012	<1%

	Publication	
13	quizlet.com Internet Source	<1%
14	www.lib.umd.edu Internet Source	<1%
15	www.tandfonline.com Internet Source	<1%
16	oenb.co.at Internet Source	<1%
17	utdr.utoledo.edu Internet Source	<1%
18	"Advanced Data Mining and Applications", Springer Science and Business Media LLC, 2017 Publication	<1%
19	C. Ackerman. "", IEEE Transactions on Robotics, 4/2005 Publication	<1%
20	www.analyticsinsight.net Internet Source	<1%
21	Hongbing Wang, Zhengping Yang, Qi Yu. "Online Reliability Prediction via Long Short Term Memory for Service-Oriented Systems", 2017 IEEE International Conference on Web Services (ICWS), 2017	<1%

Publication

22	Manas Chaturvedi. "Data mining and it's application in EDM domain", 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017 Publication	<1%
23	www.analyticsvidhya.com Internet Source	<1%
24	"Intelligent Information and Database Systems", Springer Nature, 2018 Publication	<1%
25	Chih-Chou Chiu, Deborah F. Cook, Jen-Lung Kao, Yu-Chao Chou. "Combining a neural network and a rule-based expert system for short-term load forecasting", Computers & Industrial Engineering, 1997 Publication	<1%
26	link.springer.com Internet Source	<1%
27	Xiaowei Wang, Maowei Cheng, Yefu Wang, Shaohui Liu, Zhihong Tian, Feng Jiang, Hongjun Zhang. "Obstructive sleep apnea detection using ecg-sensor with convolutional neural networks", Multimedia Tools and Applications, 2018 Publication	<1%

28	assets.cambridge.org Internet Source	<1%
29	mirrors.letterboxdelivery.org Internet Source	<1%
30	Masaharu Sakamoto, Hiroki Nakano, Kun Zhao, Taro Sekiyama. "Chapter 33 Multi-stage Neural Networks with Single-Sided Classifiers for False Positive Reduction and Its Evaluation Using Lung X-Ray CT Images", Springer Nature, 2017 Publication	<1%
31	Rohit Pathar, Abhishek Adivarekar, Arti Mishra, Anushree Deshmukh. "Human Emotion Recognition using Convolutional Neural Network in Real Time", 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), 2019 Publication	<1%
32	thinkmind.org Internet Source	<1%
33	"Artificial Intelligence in Medical Imaging", Springer Nature, 2019 Publication	<1%
34	kelsoe.com Internet Source	<1%
35	tuprints.ulb.tu-darmstadt.de Internet Source	<1%

36	Zhao Liu, Xincheng Sun, Shuai Wang, Mengjiao Pan, Yue Zhang, Zhendong Ji. "Midterm Power Load Forecasting Model Based on Kernel Principal Component Analysis and Back Propagation Neural Network with Particle Swarm Optimization", Big Data, 2019 Publication	<1%
37	Lecture Notes in Mechanical Engineering, 2016. Publication	<1%
38	Partha Pratim Barman, Abhijit Boruah. "A RNN based Approach for next word prediction in Assamese Phonetic Transcription", Procedia Computer Science, 2018 Publication	<1%
39	Nazmul Siddique. "Intelligent Control", Springer Nature, 2014 Publication	<1%
40	Chen, Ji-Long, Guo-Sheng Li, and Sheng-Jun Wu. "Assessing the potential of support vector machine for estimating daily solar radiation using sunshine duration", Energy Conversion and Management, 2013. Publication	<1%
41	vdocuments.site Internet Source	<1%
42	www.i-scholar.in	

	Internet Source	<1%
43	polen.itu.edu.tr Internet Source	<1%
44	Anish Jindal, Gagangeet Singh Aujla, Neeraj Kumar, Rajat Chaudhary, Mohammad S. Obaidat, Ilsun You. "SeDaTiVe: SDN-Enabled Deep Learning Architecture for Network Traffic Control in Vehicular Cyber-Physical Systems", IEEE Network, 2018 Publication	<1%

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		