ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

SHORT TERM ELECTRICITY LOAD FORECASTING WITH DEEP LEARNING

Ph.D. THESIS

İbrahim YAZICI

Department of Industrial Engineering

Industrial Engineering Programme

FEBRUARY 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

SHORT TERM ELECTRICITY LOAD FORECASTING WITH DEEP LEARNING

Ph.D. THESIS

İbrahim YAZICI
(507142119)

Department of Industrial Engineering

Industrial Engineering Programme

Thesis Advisor: Assis. Prof. Dr. Ömer Faruk BEYCA

FEBRUARY 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DERİN ÖĞRENME İLE KISA DÖNEMLİ ELEKTRİK YÜK TALEP TAHMİNİ

DOKTORA TEZİ

İbrahim YAZICI
(507142119)

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Tez Danışmanı: Dr. Öğr. Gör. Ömer Faruk BEYCA

ŞUBAT 2022

İbrahim YAZICI, a Ph.D. student of ITU Graduate School with student ID 507142119, successfully defended the dissertation entitled “SHORT TERM ELECTRICITY LOAD FORECASTING WITH DEEP LEARNING”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assis. Prof. Dr. Ömer Faruk BEYCA** Istanbul Technical University	....
Jury Members :	**Prof. Dr. Nizamettin BAYYURT** Istanbul Technical University	....
	Prof. Dr. Selim ZAİM Istanbul Sabahattin Zaim University	....
	Prof. Dr. Alp ÜSTÜNDAĞ Istanbul Technical University	....
	Assis. Prof. Dr. Fuat KOSANOĞLU Yalova University	....

Date of Submission : 28/01/2022
Date of Defense : 25/02/2022

To my family,

FOREWORD

In this thesis, a novel deep learning method for short-term load forecasting for a real-world case has been considered. First of all, some state-of-the-art deep learning methods and some machine learning methods for short-term electricty load forecasting with our data are investigated. According to the results, the-state-of-the-art methods outperformed conventional machine learning methods used in the thesis, hence the significance of using contemporary deep learning methods instead of using conventional machine learning methods in short-term load forecasting. Then, some variants of the used deep learning methods along with newly proposed deep learning method are deployed for the same problem. A demand side electricty load data collected between 2015 and 2017 years are used as the real-world case problem. We hope that this thesis will contribute to short-term load forecasting literature as well as deep learning literature. This study emphasizes using intelligent methods in forecasting area in so called big data regime, and provides insights about proposing novel deep learning methods for short-term load forecasting tasks into researchers.

I would like to give special thank to Scientific and Technological Research Council of Turkey (TÜBİTAK) which supported me financially during my Ph.D. under grant number 2228-B Doctorate Scholarship Programme for Master Students 2014/2.

February 2022

İbrahim YAZICI
(M.Sc.Industrial Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

AEMO	: Australian Electricity Operator Market
AI	: Artificial Intelligence
AR	: AutoRegression
ARIMA	: AutoRegressive Integrated Moving Averages
ARMA	: AutoRegressive Moving Averages
ARMAX	: AutoRegressive Moving Averages with Exogenous Variables
CART	: Classification and Regression Tree
C-LSTM	: Cycle LSTM
CNN	: Convolutional Neural Network
CRBM	: Conditional Restricted Boltzman Machine
DBN	: Deep Belief Network
DBSCAN	: Density-Based Spatial Clustering of Applications with Noise
DG	: Distributed Generation
DNN	: Deep Neural Network
DNN-CAE	: Deep Neural Network Centered Architecture Evolution
DRNN-GRU	: Deep Recurrent Neural Network Gated Recurrent Units
DRNNLSTM	: Deep Recurrent Neural Network Long-Short Term Memory
EFB	: Effective Feature Bundling
ELM	: Extreme Learning Machine
EMD	: Empirical Mode Decomposition
EPİAŞ	: Elektrik Piyasası İşletmeleri Anonim Şirketi
FC	: Fully Connected
FCRBM	: Factored Conditional Restricted Boltzman Machine
GBM	: Gradient Boosting Method
GOSS	: Gradient-based One Side Sampling
GPR	: Gaussian Process Regression
GPU	: Graphical Processing Unit
GRU	: Gated Recurrent Unit
IoT	: Internet of Things
LGBM	: Light Gradient Boosting Method
LR	: Linear Regression
LSTM	: Long-Short Term Memory
LTLF	: Long-Term Load Forecasting
MAPE	: Mean Absolute Percentage Error
MARS	: Multivariate Adaptive Regression Splines
MIMO	: Multi-Input Multi-Output
MLR	: Multiple Linear Regression
MLP	: Multi-layer Perceptron
MP	: Moore-Penrose
MPP	: Massively Parallel Processing
MSE	: Mean Squared Error
MTLF	: Medium-Term Load Forecasting
MU	: Multiplicative Unit
NN	: Neural Network

OOBE : Out-of-Bag Error
PJM : Pennsylvania, New Jersey, and Maryland
PACF : Partial Autocorrelation Function
RF : Random Forest
RMB : Residual Multiplicative Block
RNN : Recurrent Neural Network
SARIMA : Seasonal Autoregressive Integrated Moving Averages
SARIMAX : Seasonal ARIMA with External Variables
SG : Smart Grid
SIWNN : Similar day based Wavelet Neural Network
STLF : Short-Term Load Forecasting
SVM : Support Vector Machine
SVR : Support Vector Regression
TPU : Tensor Processing Unit
TD-CNN : Time Delayed Convolutional Neural Network
USA : United States of America
VI : Variable Importance
VPN : Video Pixel Network
VSTLF : Very Short-Term Load Forecasting
WT : Wavelet Transform
XGB : eXtreme Gradient Boosting

SYMBOLS

$\boldsymbol{\sigma}$	**:** Sigmoid activation function
T	**:** Hyperbolic tangent activation function
ReLU	**:** Rectified Linear Unit
*****	**:** Convolution operator
$\odot$	**:** Element-wise multiplication
$\boldsymbol{\mu}$	**:** Mean value

LIST OF TABLES

Page

LIST OF FIGURES

Page

SHORT TERM ELECTRICITY LOAD FORECASTING WITH DEEP LEARNING

SUMMARY

With the introduction of industrial revolutions, humanity has benefited from new technologies. In the third revolution, rise of electronics, telecommunications and computer systems occupied a large space. In the meantime, several learning algorithm developments began in computer science part in accordance with the rises in the third revolution. These rises and developments enabled high-level automation in the industries. In the years after the third industrial revolution, learning alogorithm developments have gone underway by researchers. Perceptrons were extended to neural networks by backpropagation introduction. As a universal function approximator, neural networks proved their efficacy in pattern recognition, function approximation, time series modelling, clustering tasks. However, depth of the networks left as a problem due to computational burdensome, lack of big data availability for training the networks, lack of enooguh powerful hardware. Support Vector Machines were also introduced for the context of classification, and extented to the context of regression later on. Several machine learning algorithms were developed in this era. These developments belonged to mainly supervised or unsupervised learning types since learning is the central theme in algorithm development efforts. On the other hand, another type of learning, that is reinforcement learning, was being deployed for sequential decision making problems by trial and error manner which is neither a fully supervised learning type nor a fully unsupervised learning type. In this type of learning, a decision making unit, especially called an agent, tries to maximize the total cumulative reward with the aim of achieving a goal in an uncertain environment by trial and error manner. Bellman operator, and Bellman optimality operator tried to yield solution ways for the problems by utilizing Markov Decision Processes in dynamic programming part. On the other hand, Q-learning, SARSA, Monte Carlo methods tried to yield solution ways for the problems in reinforcement learning part. In this type of the learning problems, very large state-spaces hinder traditional reinforcement learning algorithms to solve these learning problems. These developments and available solutions were the base for booming artificial intelligence till the first and the major breakthrough in deep neural network training came out. AlexNet introduced by a famous computer scientist Alex Krizhevsky achieved the best performance ever in the ImageNet Large Scale Visual Recognition Challenge. His work was based on training the network with many depth since it is an integral part for achieveing high performance. Hence, he created a network that incorporates convolutional layers and fully connected layers. The breakthrough came up with the introducing utilization of Graphical Processing Unit (GPU) in the training stage. The utilization of GPU in the training overcame the problem of computational burdensome of deep neural networks, and made training of them feasible. These deep networks enabled to solve the problems of traditional reinforcement learning algorithms by deep reinforcement learning. After this depth breakthrough, the depth of the networks was exploited by the researchers, thereby

boosting the performance of the networks with a great extent when compared to their counterpart machine learning algorithms. Since deep neural networks are data hunger for the training, they have benefited from the big data surge in recent times as well. Massively Parallel Processing (MPP), Graphical Processing Units utilizations for the training of deep neural networks enhanced the spread of the deployment of deep neural networks in diverse set of applications. Continuing algorithmic developments along with the big data regime, new hardware solutions supercharged the spread of the deep networks deployments. These developments in learning systems in turn contributed to the last industrial revolution that is Industry 4.0. In this revolution, transition from high-level automation extended to high-level cyber-physical systems in the industries along with technological advents. Artificial Intelligence (AI) plays important roles in this revolution, and its success is mainly based on deep neural networks in software part. Deep learning , deep reinforcement learning have been deployed for diverse set of application areas mainly including robotics, natural language processing, audio processing, computer vision, finance, healthcare, object tracking and localization, self-driving cars, however, its deployment areas are not limited to these mentioned ones.

In the big data regime and Industry 4.0, data becomes more valuable assets for the businesses. Velocity, Variety, Veracity, Value, and Volume stand for 5Vs of big data. Real-time data streaming, cloud systems, IoT, digitalization in business concepts rapidly make a stimulus for big data surge. By the time of new technological developments and algortihmic developments, increase of these 5Vs has been supercharged, creating business value from the big data has been in turn emerged as an efficient management tool for businesses. As mentioned earlier, diverse set of application areas are open for deployment of artificial intelligence techniques. In these areas, deep neural networks and deep reinforcement learning have gained importance to benefit from the data.

Energy sector is one of the beneficiary business sector that can create business value from their data available. Smart Grids (SG), Distributed Generations (DG), Energy 5.0, IoT in energy entail more intelligent techniques in part of providers since intelligent system utilization will make contributions to different levels of organizations. These contributions incorporate operational, tactical and strategical ones. Security of power supply systems, reduction in maintenance cost, efficacy of the supply system operator scheduling, reduction in penalty costs created by overestimation and/or underestimation of forecasting, market share growth, and enhancement of competitiveness among shareholders are some of these called contributions for businesses.

For the electricity provider firms, load forecasting task has a landmark importance among their operations. Beside methods to be used in forecasting, forecasting horizon is important in forecasting task since model settings are defined by regarding the horizon and the methods to be used. Forecasting horizons for electricty load forecasting are classified into four groups. These horizons are Very Short-Term Load Forecasting (VSTLF), Short-Term Load Forecasting (STLF), Medium-Term Load Forecasting (MTLF), and Long-Term Load Forecasting (LTLF). In this study, STLF is considered for the real-world case application. STLF horizon spans of half-hour-ahead up to several-day-ahaed timesteps. Energy market establishments have been developed by introducing market regulations in Turkey since 2001. After many regulations and transitions from state-run-market to a non-governmental regulated market, Energy Markets Enterprise Corporation (EPİAŞ in Turkish)was established in 2015. In this market, the day-ahead-market, intraday market and balancing market mechanisms play important roles for the electricity system management in Turkey. These mechanisms plays complementary roles for each other.

In this market, stakeholders aim to avoid extra costs arose in balancing market where deficient and excessive amounts of electricity are compensated by purchase and sale among stakeholders since the market imposes 3% penalty costs for these deficient and excessive amounts. And this avoidance can be facilitated by efficient forecasting performance. Hence, forecasting task arises as an important tool for decision makers in forecasting. Before transtion to a regulated market by EPİAŞ, predctions are performed mainly weekly or more than one-week-ahead. The error margins for the predictions made were in turn very high and flexible. This flexibility provided the electricty providers in the market to compromise their excessive and deficient amounts easily when compared to the regılated market situations. Flexibility in the prediction error margins enabled the providers to meet their requirements in the market in the ong horizon with less price charge. In the regulated market, the day-ahead market and intraday market mechanism have turned out to be the integral part of the market mechanisms. Sustaining the competition in the market, growing the market share, reducing the operational costs, and penalty costs created by overestimation and underestimation of the load forecasting, tasks of one-hour-ahead forecasting and one-day-ahead forecasting located at the heart of major concerns for the electricity provider firms in the regulated market. The provider firms in turn focused on these tasks to achieve the aforementioned goals, then create business value through performing these tasks. Thus, this study focuses on the major concerns for the providers by deploying deep learning algorithms for a real-world case.

In this study, electricty load data which consists of hourly load demands, for 3 years collected between 2015 and 2017 years were utilized. The granularity of the time series data obtained was composed of load values and temperature values as it is used for the regular forecasting task by the provider firm. In the first stage of applications, preliminary data examinations were performed which provides a guide for time series problem handling for both applications of conventional machine learning, and deep learning methods. These data examinations contained data normalization, dummy variable inclusion, autocorrelation identification tasks for each method type. This stage is followed by input set preparation for the methods deployed. We framed our dataset into a supervised learning dataset by shifting values according to the results of autocorrelation identification, that is time lag. Weekly time lag was found the best choice, hence we used this time lag value in our framing. In addition, since neural networks are at the heart of the applications in this study, we used data normalization as zero-mean normalization to facilitate fast convergence and numerical stability for the networks in training and testing. After preliminary data examinations, we conducted comprehensive comparative analyses of the methods. In the first round of the comparative analyses, two deep learning methods and some popular machine learning methosds were compared whether deep learning methods overcome the conventional methods in STLF task. The deep learning methods were in turn found superior to the conventional methods used which the results were validated by statistical significant test. In the second round of the comparative analyses, just deep learning methods were compared. This round of the comparisons was the central theme in this study since the aim was to propose a deep learning method for the real-world case. For this reason, we proposed a new method based on one-dimensional convolutional neural networks, and compared its performance with the other deep learning methods by applying them to the real-world case. As per the results obtained from this round of comparisons, the proposed method proved its efficiency for both one-hour-ahead, and one-day-ahead prediction tasks. This fact was also validated by statistical significance test as well.

In brief of this study, there are some level of takeaways from the results of the study. At the organizational level takeaways, intelligent technqiues use especially in energy sector such as deep learning, deep reinforcement learning tools will make contributions to organizations with different levels. Secondly, energy sector is one of the businesses that enormous amount of data is hoarded even hourly. Hence, creating business value by utilizing intelligent systems in their operations will enable short-term, mid-term, and long-term achievements for them when considered big data regime, advents in hardware and software solutions, and developments in artificial intelligence methods especially in neural networks. At the most conceptual level, deep learning methods provide high-performance forecasting engine for the providers for STLF as per the results obtained. Deployment of these type of artificial intelligence method will make them at the front line in the market. At the method-level takeaways, calendar effects have landmark importance in time series modelling for STLF. Rare time issues, and dual calendar effects are another landmark important issues in time series modelling as well. Efficient feature extraction ability of Convolutional Neural Networks (CNN), and auto-capturing long-term relations in long sequences make them a rival for Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) in time series modelling tasks besides the tasks of audio recognition, speech recognition, natural language processing. In addition, the proposed method’s exogenous variable inclusion for modelling the time series problems boosts the performance of the method since different level of resolutions are captured by this setting. Hence, this setting can be extended for later method developments of deep learning methods.

DERİN ÖĞRENME İLE KISA DÖNEMLİ ELEKTRİK YÜK TALEP TAHMİNİ

ÖZET

Endüstriyel devrimlerin ortaya çıkmasıyla birlikte, insanlık geliştirilen yeni teknolojilerden fayda sağlamaktadır. Üçüncü endüstri devriminde, elektronik, telekomünikasyon ve bilgisayar sistemlerindeki artış çok büyük bir yere sahiptir. Aynı zamanda, bilgisayar bilimleri alanında çeşitli öğrenme algoritmalarının geliştirilmesi yine üçüncü endüstri devrimindeki ortaya çıkan ilerlemelere paralel olarak başlayıp devam etmiştir. Bu ilerlemeler endüstrilerde yüksek düzeyli otomasyon sistemine geçişi sağlamıştır. Üçüncü endüstri devrimi sonrası algoritma düzeyindeki ilerlemelerde, yapay sinir ağlarının temelini oluşturan perseptron algoritması geri yayılımlı eğitim algoritması sayseinde geliştirilerek aynı katmanda çok perseptronlu yapay sinir ağlarına dönüştürülmüştür. Yapay sinir ağları genel bir fonksiyon yakınsaması alogritması olarak örüntü tanıma, fonksiyon uydurma ve yakınsama, zaman serisi modelleme ve kümeleme analizleri alanlarında etkinliğini kanıtlamıştır. Ancak derin yapay sinir ağlarının eğitimi için gerekli büyük veri, hesaplama güçlüğü, hesaplamalar ve eğitim için gerekli güçlü donanım eksikliği yapay sinir ağlarındaki derinlik meselesini problem olarak bırakmıştır. Destek vektör makineleri de ayrıca sınıflandırma amaçlı önerilen algoritmalardan olup, daha sonra regresyon amaçlı versiyonu da literatüre kazandırılmıştır. Sayılan iki önemli algoritma dışında çeşitli makine öğrenmesi algoritmaları yine bu dönemde geliştirilmiştir. Bu geliştirmelerin çoğu algoritma geliştirmedeki ana amaç öğrenme olduğu için gözetimli ve gözetimsiz öğrenme alanlarına aittir. Bunun yanında üçüncü bir öğrenme tipi olan sıralı verilerin olduğu durumlarda karar verme yöntemi olan pekiştirmeli öğrenme tipi de, ne tam olarak gözetimli ne de tam olarak gözetimsiz öğrenme sınıfına düşen, deneme yanılma metodolojisine dayalı olarak kullanılan bir yapay zeka metodları sınıfıdır. Pekiştirmeli öğrenme metodolojisinde, ajan ismi verilen bir karar verme ünitesi, deneme yanılma yoluyla etkileşimde bulunduğu belirsiz çevrenin dinamiklerine göre toplam kümülatif ödül değerini maksimize etmeyi amaçlamaktadır. Bellman operatörü ve Bellman optimallik operatörü dinamik programlama perspektifinden Markov Karar Süreçlerini kullanarak pekiştirmeli öğrenme problemleri için çözüm yolları üretmeye çalışmıştır. Bunların yanında, Q-öğrenmesi, SARSA, Monte Carlo metodları pekiştirmeli öğrenme perspektifinden pekiştirmeli öğrenme problemleri için çözüm yolları üretmeye çalışmıştır. Pekiştirmeli öğrenme tipinde büyük uzay-durum kümeleri klasik pekiştirmeli öğrenme algoritmalarının bu problem tiplerinin çözümünde etkinliğini kısıtlamaktadır. Anlatılan gelişmeler ve mevcut çözümler yapay zekanın ilk ve en büyük atılımından önceki en önemli etkenlerdi. Ünlü bir bilgisayar bilimcisi olan Alex Krizhevsky tarafından önerilen AlexNet modeli 2012 yılında ImageNet Büyük Ölçekli Görsel Tanımlama Yarışması'nda tüm zamanların en iyi performansını gerçekleştirmiştir. Çalışmasının temeli, yapay sinir ağlarının yüksek performansını sağlayan en önemli etken olan yapay sinir ağlarındaki derinliğin artırılmasına

dayanmaktadır. Bundan dolayı Alex Krizhevsky evrişimli katmanlar ve full bağımlı katmanları içeren bir model olan AlexNet'i önermişlerdir. AlexNet'in büyük başarısı önerilen yapay sinir ağı modelinin grafik işlemcileri üzerinde eğitilmesiyle sağlanmıştır. Grafik işlemcilerinin kullanılması derin yapay sinir ağlarının eğitilmesindeki hesaplama gücü zorluğunu kolaylşatırmış ve onların bu şekilde eğitimini mümkün hale getirmiştir. Üretilen bu derin yapay sinir ağları klasik pekiştirmeli öğrenme algoritmalarıyla çözülemeyen problemler için de çözüm olarak ortaya çıkmıştır. Bu derinlik devriminden sonra araştırmacılar yapay sinir ağlarının derinliği konusunda arşatırmaları artırarak, bu şekilde yapay sinir ağlarının performansını diğer klasik makine öğrenmesi algoritmalarına oranla daha da artırmışlardır. Derin yapay sinir ağları veri açısından büyük çaplı veriye ihtiyaç duyduğundan son zamanlardaki trend olan büyük veri artışı durumundan çok fazla fayda sağlamıştır. Derin yapay sinir ağları eğitiminde yoğun paralel işleme üniteleri, grafik işleme üniteleri kullanımları çok çeşitli uygulama alanalarında bu ağların kullanımını yaygınlaştırmıştır. Derin yapay sinir ağlarının hızlı bir şekilde yükselişe geçmesi büyük veri rejiminin hzılı yükselişi, buna paralel olarak algoritmik gelişmelerin de devam etmesi ve yeni donanımsal çözümlerin geliştirilmesi sonucunda oluşmuştur. Bunun sonucunda ise son endüstri devrimi olan Endüstri 4.0 bu öğrenme sistemlerinin gelişmesi ile katkı sağlanmıştır. Bu devrimde yüksek seviyeli otomasyondan daha da ileri nokta olan siber fiziksel sistemlere geçiş bu yeni gelişmeler temelli teknolojilerle sağlanmıştır. Yapay zeka bu devrimde çok büyük bir yere sahiptir ve de bu öenmli konuma sahip olma etkinliği yazılımsal açısından derin yapay sinir ağlarına bağımlıdır. Derin öğrenme, derin pekiştirmeli öğrenme robotik, doğal dil işleme, ses işleme, bilgisayarla görü, finans, sağlık, nesne takibi ve konumlandırma, otonom araç gibi alanlarda başta olmak üzere çeşitli alanlarda uygulama sahalarına sahiptir.

Büyük veri rejimi ve Endüstri 4.0 çağında veri işletmeler açıısndan çok değerli bir varlık olmaktadır. Hız, çeşitlilik, değer, hacim ve doğruluk büyük verinin 5 özelliğindendir. Gerçek zamanlı veri akışı, bulut sistemleri, nesnelerin intenreti, dijitaleşme gibi konseptler büyük veri yükselişine ivme katmaktadır. Yeni algoritma ve teknoloji geliştirmeleri, büyük datanın sayılan özelliklerindeki nitelik artışını çok hızlı bir şekilde ivemlendirmiş ve bunun sonuncunda büyük veriden iş değeri üretmek şirketler açısından önemli bir yönetim aracı haline gelmiştir. Daha önce de belirtildiği gibi yapay zeka uygulamaları için çeşitli uygulama alanları mevcuttur. Bu uygulama alanlarında derin yapay sinir ağları ve derin pekiştirmeli öğrenme konuları veriden değer üretme konusunda önem kazanmaya devam etmektedir.

Enerji sektörü elindeki mevcut veriler sayesinde bunların iş değerine döünüştürülebileceği ve bu şekilde büyük veriden fayda sağlanabilecek alanlardan birisidir. Akıllı şebekeler, dağıtık sistemler, Enerji 5.0, enerjide nesnelerin interneti gibi konseptler elektrik sağlayıcıları açısından daha akıllı teknolojilerin kullanılmasını gerektirmektedir çünkü akıllı teknolojilerin kullanılması organizasyonlara çeşitli seviyelerde katkılar sağlamaktadır. Bu katkılar operasyonel, taktiksel ve stratejik katkıları içermektedir. Enerji tedarik sisteminin emniyeti ve stabilizasyonu, bakım maliyetlerinde azalma, tedarik sistemi operatörü çizelgeleme etkinliğinin artışı, tahmin hataları kaynaklı maliyetlerde azalma, pazar payı artışı, pazarda rakiplere karşı rekabetçiliğin artışı vb. Katkılar bahsi geçen katkı türlerinin bir kısmıdır.

Elektrik üretici firmaları için elektrik talep tahmini operasyonları içinde hayati bir yere sahiptir. Talep tahmininde kullanılan metodların yanında, tahmin ufku da önemli bir yere sahiptir. Tahmin ufkuna göre metod seçimi yapılması ve girdi değişkenleri ayarlanması yapılması tahmin ufkunun önemini vurgulamaktadır. Elektrik talep tahmini konusunda tahmin ufku dört ana grupta incelenmektedir. Çok kısa dönemli, kısa dönemli, orta dönemli ve uzun dönemli talep tahmini ufukları bu dört grubu oluşturmaktadır. Bu çalışmaya konu olan problem gerçek bir kısa dönemli talep tahminidir. Kısa dönemli talep tahminin ufku bir saatten birkaç gün öteye kadar olabilmektedir. Enerji piyasası düzenlemeleri Türkiye'de piyasa yasal düzenlemelerinin başladığı 2001 yılından itibaren süregelmektedir. Devlet kontrolündeki piyasadan serbest piyasaya geçişteki yasal düzenlemeler ve uygulamalardan sonra Enerji Piyasaları İşletmeleri Anonim Şirketi (EPİAŞ) 2015 yılında kuruldu. Bu piyasada gün öncesi, gün içi ve dengeleme piyasaları mekanizmaları Türkiye'deki elektrik sistemi yönetimi açısından önemli rol oynamaktadır. Bu mekanizmalar birbirlerini tamamlayıcı roller de oynamaktadırlar.

Bu piyasada paydaşlar eksik ya da fazla elektrik üretiminden kaynaklanan farkların paydaşlar arasında al-sat mekanizmasıyla tazmin edildiği dengeleme piyasasında fazla maliyetlerden kaçınmak istemektedirler. Bunun nedeni gereğinden düşük ya da fazla olan miktarlar dolayısıyla %3 oranında ceza maliyeti ödemektir. Bu ceza maliyetinden kaçınma etkin bir tahminleme ile mümkün olabilmektedir. Bu nedenlerden dolayı tahminleme işi bu alandaki yöneticiler için önemli bir araç olmaktadır. EPİAŞ'ın yönettiği piyasadan önceki dönemde tahminlemeler haftalık ya da birkaç hafta ötesinden gerçekleştirilmekteydi. Bundan dolayı da hata marjları yüksek ve esnekti. Bu esneklik elektrik tedarikçilerine gereğinden fazla ya da düşük olarak ürettikleri miktarı dengelemek için yeni düzenlenen piyasa şartları düşünüldüğünde kolaylıklar sağlıyordu. Hata marjlarındaki esneklikler tedarikçilere uzun vadede düşük maliyetlerle piyasadaki teminatlarını gerçekleştirmeye imkan veriyordu. Yasal düzenlemeler sonucu oluşan piyasada gün öncesi ve gün içi piyasaları bu piyasanın vazgeçilmez parçaları olmuştur. Piyasadaki rekabetçiliğin sürdürülmesi, pazar payı artışı, operasyonel giderlerin azaltılması, tahmin hataları sebebiyle ortaya çıkan ceza maliyetlerinin düşürülmesi, bir saat sonrası ve bir gün sonrası tahminlemelerin bu piyasada merkezi konuma gelmesini sağlamıştır. Bunun sonucunda bu tahminleme operasyonları tedarikçiler açısından odaklanılan bir alan olmuş ve bu operasyonlarla veriden iş değeri elde etme işlemlerine de imkan sağlamıştır. Bu değerlendirmeler ışığında teze konu olan çalışma tedarikçiler için önem arz eden kısa dönemli elektrik talep tahmini konusunda gerçek bir tahminleme problemi için derin öğrenme tekniklerinin uygulanmasını içermektedir.

Bu çalışmada İstanbul'da bir bölgede 2015-2017 yıllarına ait üç yıllık elektrik tüketim verisi kullanılmıştır. Elde edilen zaman serisi verisi içerik olarak ilgili saatlere ait tüketim ve sıcaklık değerleri ile birlikte gün ve saat değerlerini içermektedir. Bu özellikler ayrıca ilgili tedarikçi firma tarafından da tahminleme işlemleri için kullanılmaktadır. Uygulamaların ilk aşamasında hem derin öğrenme hem de makine öğrenmesi yöntemlerinin zaman serileri problemlerine uyarlanması dikkate alınarak yönerge halinde veri inceleme çalışmaları gerçekleştirilmiştir. Bu incelemeler her bir metod için veri normalizasyonu, kategorik veri eklenmesi, otokorelasyon belirlenmesi çalışmalarıdır. Bu aşama uygulanan yöntemler için girdi seti oluşturmayı takip etmiştir. Zaman serisi verisi otokorelasyon incelemesi sonucu belirlenen geri bakma süresine göre veriler otokorelatif şekilde geriye doğru ötelenerek veri seti gözetimli öğenme veri setine çevrilmiştir. Bunun yanında tezin odak noktasında yapay sinir ağları olduğundan dolayı hızlı yakınsama ve sayısal stabilizasyon sağlamak adına sıfır ortlamalı normalizasyon yöntemi kullanılmıştır. İlk aşama veri incelemeleri sonrasında ilgili yöntemlerin geniş kapsamlı karşılaştırılmaları gerçekleştirilmiştir. İlk karşılaştırma çalışmasında zaman serileri için popüler makine öğrenmesi metodları ve iki derin yapay sinir ağı modeli karşılaştırılmıştır. Bu karşılaştırmada derin öğrenme metodlarının klasik makine öğrenmesi metodlarına uygulmaya konu olan verinin kısa dönemli talep tahmininde üstünlük sağlayıp sağlayamadığı belirlenmeye çalışılmıştır. Bu karşılaştırmaların sonucunda istatistiksel testlerin de gerçeklemiş olduğu sonuçlara göre derin yapay sinir ağları uygulanan klasik makine öğrenme metodlarından daha iyi sonuçlar üretmiştir. İkinci karşılaştırmada ise sadece derin yapay sinir ağları modelleri karşılaştırılmış olup tezin odak noktası bu karşılaştırmalardan oluşmaktadır. Çünkü tezin amacı gerçek bir problem için yeni bir derin yapay sinir ağı metodu önermeyi amaçlamaktadır. Bu amaçlar dolayısıyla bu tez çalışmasında bir boyutlu evrişimli sinir ağlarına dayalı bir metod ilgili problem ele alınarak önerilmiş ve de bu metodun performansı diğer derin yapay sinir ağları metodlarının performansları ile karşılaştırılmıştır. İkinci karşılaştırmalar sonucunda elde edilen sonuçlara göre bir saat sonrası ve bir gün sonrası talep tahmini işlemlerinde önerilen metod etkinliğini kanıtlamış olup bu durum istatistiksel önem testleriyle de desteklenmiştir.

Tezin kısa özeti olarak, bu çalışmadan çıkarılacak farklı düzeyde çıkarımlar bulunmaktadır. Organizasyonel düzeyde bu çıkarımlara bakılırsa, derin yapay sinir ağları ve derin pekiştirmeli öğrenme gibi akıllı metodların özellikle enerji sektöründe kullanılması organizasyonlara taktiki operasyonel ve stratejik seviyelerde farklı katkılar sağlayacaktır. İkinci olarak ise enerji sektörü anlık verilerin toplandığı sektörlerden birisidir. Büyük veri rejimi, yazılımsal ve donanımsal çözümlerin gelişmesi, yeni derin yapay sinir ağları modelleri geliştirilmesi konuları göz önünde bulundurulduğunda kısa, orta ve uzun vadeli başarıların kazanılması için ismi geçen akıllı yöntemlerin kullanılmasıyla veriden iş değeri üretilmesi konusu önemli bir yer tutmaktadır. Kavramsal düzeyde ise elde edilen sonuçlar neticesinde kısa dönemli talep tahminleri için derin yapay sinir ağları yüksek performans üreten tahminleme araçları olmaktadır. Bu tip yapay zeka metodlarının uygulanması onların piyasada ön plana çıkmasını sağlamıştır. Metod düzeyinde ise takvimsel etkilerin kısa dönemli zaman serisi modellenmesinde çok önemli bir yere sahip olduğu görülmüştür. Ayrıca nadir zamanlarda kaynaklanan durumlar ve çift takvim etkileri de kısa dönemli zaman serisi problemlerinin modellenmesinde önemli bir yer tutmakta olduğu görülmüştür. Evrişimli yapay sinir ağlarının etkin özellik çıkarma özelliği ve verideki uzun dönemli ilişkileri otomatik yakalama kabiliyetleri, evirşimli yapay sinir ağlarının ses tanıma, doğal dil işleme işlemlerinin yanı sıra zaman serisi problemleri için de bu ağların LSTM ve GRU karşısında rakip olarak ortaya çıkabileceğini göstermektedir. Dışsal veri ekleme kabiliyetine sahip olan önerilen metod bu sayede diğer derin öğrenme metodlarına göre daha iyi sonuç üretmiştir. Bunun nedeni ise evrişimli sinir ağlarına uygulanan veri girdisi ayarlarının veri içindeki farklı düzeydeki örüntüleri etkin şekilde tanıması olarak gösterilebilir. Bundan dolayı bu ayarlar daha sonraki derin yapay sinir ağları metodları geliştirilmeleri için kullanılabilir.

1. INTRODUCTION

Artificial Intelligence (AI) is considered the new electricty nowadays when considered Industry 4.0 concept. It shapes our world again as the previous advancements did before, hence seems to transform every industry by creating economic value. Big data surge in recent years, software and hardware advancements such as Massively Parallel Processing (MPPs), Tensor Processing Units (TPUs), Graphical Processing Units (GPUs), and novel algorithm developments for neural networks contribute to the emergence of Artificial Intelligence. Coincidence of big data surge and advancement of data hunger deep learning algorithms supercharged the widespread of artificial intelligence and its impacts accross industries. Healthcare [1], computer vision [2], fault detection and identifiction in sensory systems [3], human activity recognition [4], natural language processing [5], robotics [6], autonomous driving [7], marketing and advertising [8], finance [9], recommendation systems [10] are some exemplary studies of artificial intelligence applications in different industries.

AI techniques mainly work by extracting patterns in the enormous amount of data.In this extraction task, learning manner of algorithms used determines the type of learning. Supervised and unsupervised learning are the types of learning used for AI.Supervised learning scheme works by knowing in advance what to look for, namely it knows the labels of the problem considered. On the other hand, unsupervised learning scheme works-without knowing what to look for in advance and try out to propose an intuitive result for underlying pattern hidden in data observed. Another scheme that is one of the integral part of AI techniques is reinforcement learning which uses netiher supervised nor unsupervised learning scheme, but instead uses trial and error scheme for the pattern extraction in data. Support Vector Machines (SVMs), Neural Networks (NNs), k-means clustering, Logistic Regression, Lasso Regression and so forth are the basic examples of AI methods. After the aforementioned advancements in data amount, hardware and software, and algorithms, neural networks benefited extensively from those. While there were no clear performance differences among conventional machine learning algorithms and neural networks, deepening the neural networks and short processing time for training them with massive data in shorter time with respect to old era supercharged the performance of these networks, hence paved the way for outperformance of the novel deep neural networks with respect to traditional machine learning algorithms. Long-Short Term Memory (LSTM), Gated Recurrent Units (GRU), Convolutional Neural Networks (CNN), Transformers are the novel neural network types that use deep architectures in their layers. As mentioned before,these novel deep networks are prevalently used across many areas such as natural language processing, healthcare, recommender systems, robotics, autonomous vehicles, computer vision, image processing, human activity recognition, visual tracking and so forth. One of the areas that can benefit from the novel deep networks is electricity load forecasting area in which several different statistical and machine learning algorithms are already in use. Decision makers and managers in electricity provider firms may benefit from the intelligent methods emerged instead of conventional machine learning methods and statistical methods.

For the forecasting task of electricity load, forecasting horizons are classified into four categories as per their forecast steps. Very Short-Term Load Forecasting (VSTLF), Short-Term Load Forecasting (STLF), Medium-Term Load Forecasting (MTLF), and Long-Term Load Forecasting (LTLF) are these mentioned four forecasting horizons.

This study aims to provide a novel deep learning method for Short-Term Load Forecasting for an electricity provider firm in Istanbul, Turkey. To pursuit the goal of proposing a novel deep learning method that can challenge its existent deep learning counterparts in the literature, several comparative studies between machine learning and deep learning methods, and within deep learning methods are made through utilizing a real-world application data. In the first stage of deploying machine learning methods and some of deep learning methods, three different deep learning methods and seven different machine learning methods, which are frequently used for STLF, are compared, and it was found that deep learning methods outperformed the rest of the methods deployed. In the second phase of the methods' deployment, a novel method, one-dimensional Convolutional Neural Networks based on Video Pixel Networks (VPNs), and some other existent deep learning methods and their variants are compared. As the real world-case application in this thesis, demand-side electricity load data collected between the years 2015 and 2017 in Istanbul are used.

The organization of the thesis is as follows; in the next sub-sections of the first chapter of this study, STLF and its integral parts in relation to Turkish electricity market is introduced.This sub-section was followed by motivation behind the study, purpose of the thesis, and unique aspects of the study sections. In the second chapter, the literature contemporary literature of deployments of deep learning methods for STLF is combed, similar and different sides of the extant literature with this study are reviewed. In the third chapter, background of the methods used in the comparative studies, STLF data preparation and model settings are extensively presented, and discussions of the results of the comparative studies are made. In the final chapter of the study, concluding remarks and future research directions are given.

1.1 Short-Term Electricity Load Forecasting

Electricity load forecasting has great importance for different components in the market which consists of reasearchers, producers, providers firms, public adminstrators, contarctors [11]–[13]. Load forecasting issue has different levels of impacts for power suppliers in the market, and these impacts include operation, tactical and stratgeical level impacts. Operational cost management, sustainability and security of the power supply systems are significant impacts of efficient load forecasting in the short and mid-term [14]–[16]. On the other hand, market share growth, increase in the competetion of the firms are some significant impacts of the forecasting in the long term [17]. Several reasons are behind the developments of novel forecasting methods, and forecasting engines to obtain more accurate results than the current ones, in turn to create business value from data through using the developments. Transitions to regulated electricity markets [12], [18], more efficient and flexible forecasting engines or methods due to big data availability in the electricity markets [19], [20], intelligence requirements in the electricity markets due to new technologies such as smart grids, distributed generations, Internet of Things (IoT)[21], [22] may be considered the main reasons behind the novel methods developments. Many researchers study on the forecasting area, and big data surge, advents in hardware solutions, and developments in neural networks by deep architectures have lent impetus for introducing novel methods in this area as well, and the primary aim of this thesis is relevant to this mentioned fact that is to propose a novel deep learning method for STLF task of a provider firm.

As mentioned before, the forecasting horizons are grouped into four classes. VSTLF timesepan is between a few minutes and one hour [23], MTLF timespan is between several weeks and several months while LTLF timespan is between one year and several years [24]. On the other hand, STLF timespan is between one hour and one week. Even diffeent forecasting horizons are interrelated, they have different impacts and contributions for different management levels for the provider firms. LTLF makes contributions in strategic level, MTLF makes contributions to sustainability of the power supply systems. STLF contributes to planning and management of the power supply systems for the providers. Moreover, since the day-ahead and intraday electricity prices are spotted by STLF, it turns out to be the integral part of managerial decision makers of the providers in forecasting tasks [25].

1.2 Motivation

Dynamics of the energy markets of the countries may vary due to regulations and policies in the countries. After transitions to regulated markets, the provider firms adjust themselves to newly introduced regulations and polices through making streamlines of their operations. In Turkey, the first step of changing the dynamics of the energy markets started by introducing regulations for the market in 2001. Since then, many regulations have been put in place. In Turkish energy markets, establishments in the energy market including day-ahead-market planning, financial consolidation and hourly pricing were put in place in 2009 as per the regulations. This was followed by down-payment and assurance, incentives for enforce renewable energy programs were established after two years. In 2013, legislation for foundation of Energy Market Enterprise Corporation (EPİAŞ) was done, and EPİAŞ has been ruling the electricity market since 2015. In the same year, intraday market mechanism for electricty was introduced. In the electricity market regulated by EPİAŞ, day-ahead, intraday and balancing market mechanisms are the integral part of operating the market. Before transitioning to a regulated market by EPİAŞ, prediction horizons for the provider firms were very flexible which can vary from one-week-ahead to several months ahead, hence these flexible ranges were providing large error margins in the tasks of forcasting for the provider firms when the regulated market conditions are considered. Hence, balancing the excessive or deficient amounts of the electricity with less price was the result of the market before transtioning to the regulated market. Day-ahead and intraday market mechanisms introduced by EPİAŞ started to restrict the aforementioned flexibility in the forecasting horizons for the provider firms, then cost management and competitiveness in turn gained landmark importance in the market. In these two market mechanisms, one-hour-ahead prediction task is the integral part of intraday market. On the other hand, one-day-ahead prediction task is the integral part of the day-ahead market mechanism. Since day-ahead market and intraday market mechanisms along with balancing market mechanism interact and complement each other, reducing the costs caused by overestimation/underestimation depends on efficiently manage these mechanisms for STLF. And, this efficient management depends on accurate predictions in STLF where one-hour-ahead and one-day-ahead predictions are the most important subjects [26].

Importance of short term load forecasting has been the subject of many studies. The significance of short term load forecasting which contributes to security and sustainability of the power supply system, reduction in operational costs, efficient scheduling and management of the power supply systems is emphasized by many studies in the extant literature [18], [27]–[31]. These mentioned achievements can be provided by accurate forecasting task. Due to the tarnsformations in all areas made by AI boosting, technological advancements transforms the electricty providing sector as well. Introduction of smart grids (SGs), distibuted generations (DGs), Smart Grid Plus, Energy 5.0, the Energy Internet, the IoT concepts entail the use of more intelligent systems in the electricty providing sector [21]. This entailment will in turn help the providers primarily create business value besides contributing to different levels of management levels [22], [29], [30]. On the other hand, use of the intelligent methods provided by new algorithm developments in energy-related problems has been gaining huge importance recently [31]–[33]. While conventional machine learning methods and statistical learning methods may provide acceptable forecasting results for the forecasting tasks, deep neural networks supersede conventional machine learning methods and statistical learning methods in this area by taking advantage of coincided with big data surge, technological advancements, and broadly use of the intelligent methods in the load forecasting area.

The motivation behind this study stems from above mentioned facts which are the major concerns for the providers in STLF in the newly regulated market that are one-hour-ahead and one-day-ahead predictions, and emergence of deployments and proposing intelligent methods for STLF which are majorly based on deep neural networks.

1.3 Purpose of the Thesis

AI has been gaining huge importance in recent years, and transforming the industries as well as the way of our ordinary lives. Aforementioned big data surge, advents in hardware and software solutions, developments of novel algorithms based on deep neural networks boosted AI spread across the businesses. One of the businesses that can benefit from AI spread is electricity providing sector in which regulated market transitions in many countries, and the requirement of the use of intelligent systems for accurate predictions that will in turn create business values beside helping the providers manage the power supply systems securely and more effieciently. By considering these facts, both theoretical and practical advents of deep neural networks emerge in many areas due to their outperformance over conventional methods since massive amounts of data, new hardware and software solutions work complementarily with them by supercharging their performances. Long-Short Term Memory, Gated Recurrent Units, and their variants are prevalently used for STLF tasks. Convolutional Neural Networks has proved their efficacies in many tasks such as natural language processing, audio recognition, image processing, computer vision, object tracking, human activity recognition. Recently, CNN use has emerged in STLF tasks as well. By considering intelligent method development and deployment in STLF tasks, we aimed to propose a novel method based on CNN that can be used for STLF task for a provider firm in Turkish electricty market, and challenge with the extant deep learning methods such as LSTM, GRU, and their variants. A real-world case problem is considered by developing the novel method. Demand-side electricity load data which consist of hourly temperature and load collected between 2015 and 2017 in Istanbul, Turkey are used in this thesis.

Reducing the forecasting error of the provider firm by introducing an intelligent method is the primary aim of this thesis. In addition, proposing a novel intelligent method based on deep learning is the another aim of this thesis.

1.4 Unique Aspects of the Study

In the method deployment stage of this thesis, firstly, ten methods including some of deep learning methods and machine learning methods frequenty used in STLF tasks were compared for the tasks of one-hour-ahead forecasting and one-day-ahead forecasting. This first comparative analysis of the methods showed the superiority of deep learning methods over conventional machine learning methods for data used. Secondly, main comparative analysis was conducted by deploying LSTM, GRU, some of their variants and newly proposed one-dimensional CNN method based on Video Pixel Networks (VPNs). We aimed to make some contributions to STLF area with this study at the most conceptual level;

- For sequential data, 1-D CNN use is rare when LSTM, GRU, and their variants use in this type of data. Even 1-D CNN use has appeared in recent studies, to the best of our knowledge, 1-D CNN based on VPNs method for STLF is not available in the extant literature. Hence, this study deploys a novel method used with three-dimensional video data for one-dimensional sequential data of time series for STLF,
- The proposed 1-D CNN based on VPNs is used by some adjustments in gating mechanisms of VPNs. We don’t only deploy VPNs based CNN method, also make some modifications in its gates to bring a novelty in this thesis. In this sense, our modifications produce promising results discussed in the later sections of the thesis,
- The comparative studies provide insights about data processing, model deployement, and feature selection in STLF area into the manegarial decision-makers in this area by their results especially through deep learning method deployments [26].

2. RELATED WORKS AND LITERATURE REVIEW

Application of deep learning methods in the extant literature are reviewed under two settings as the applications of their machine learning counterparts. One of these settings is the application of single method that the method solely is a deep learning method. On the other hand, the second setting consists of a hybrid method application of deep learning methods, machine learning methods and/or decomposition methods. The single method applications is intuitive for both of the practitioners and the researchers in STLF area, and provide off-the-shelf application methods while the hybridized method applications provide promising results as the single method applications though they require extensive domain knowledge, method-level knowledge. Moreover, they may not exploit the nature of deep neural networks efficient pattern extraction in data by correlation spotting by CNN, recurrency in RNNs, forget and update gates in RNN. However, there is no an absolute solution whether to use single method application settting or hybrid method application one. The existent studies in the literature on STLF support the mentioned fact, and both of these two settings are prevalently encountered ones in the literature.

In this part of thesis, we combed the literature on short-term electricity load forecasting by deep learning methods, discussed the implementations of the methods and nexus of them with our study.We give the related works as per the used deep learning methods in sub-sections of the literature review.

2.1 RNN, GRU and LSTM in STLF

Recurrent Neural Networks and its variants, LSTM and GRU, are good at processing sequential data that exhibit trend, seasonality, time dependency and cycles. Since STLF task contains sequential data to process, both single method application or hybrid method application settings of RNN and its variants are extant in the literature. [34] used three inference approaches for forcasting that are direct, recursive and multi-input multi-output ones. The authors applied different models by considering these approaches. The models are only use of recurrent blocks, use of bi-directional recurrent blocks, and recurrent and bi-directional recurrent block use on top of a 1-D CNN for STLF. [35] used both of one-layer RNN and a deeper RNN to benefit from the potential of efficacy of deep networks. They performed prediction for different levels of power systems which one of them is New England system's regionally aggregated load and data of 100 Irish disaggregated households. [30] combined attention mechanism and rolloing update with bi-LSTM in their study. The attention mechanism used for reflecion of spatial importance of historical data in different time intervals. The rolling update with bi-LSTM used to improve the prediction accuracy to simultaneously providing past and future information available in data used. Australian electricity load data set was used to prove the efficacy of the combined method. [36] dealt with online prediction task for smart meter data. The authors proposed an approach by RNN which is able to adaptively tune its parameters for streaming data. They tested the proposed approach with several other methods with five different data sets. According to the results obtained by the proposed approach, not only higher accuracy was obtained but also computation time advantage gain for this type of data was shown. [37] aimed to show efficacy of uses of LSTM and GRU methods for STLF. GRU method was found superior to LSTM with used load data. [38] proposed a GRU based deep RNN (DRNN-GRU), and applied it for one-hour-ahead prediction task. Both aggregated and disaggregated demands of residential hourly load data in Texas, USA were used to show efficacy of the proposed method. The proposed method was compared with DRNN-LSTM, simple RNN, multi-layer perceptron (MLP), Autoregressive Moving Average (ARIMA), Support Vector Machine (SVM) and Multiple Linear Regression (MLR).

[39] used LSTM method for STLF with 10-year electricity load data belonging to Ontario in Canada. LSTM method yielded better results than shallow NN and ARIMA as per the obtained results. [40] used an LSTM method based on attention mechanism, and applied the method for six-timestep-ahead forecasting task. The method was compared with several other machine learning methods and deep learning methods. [41] used GRU and LSTM methods for STLF task, and GRU method produced better results than LSTM method with data applied. [42] used LSTM method for the tasks of one-day-ahead, two-day-ahead, one-week-ahead, and one-month-ahead forecasting. The authors compared the method with Autoregressive Moving Average (ARMA), Seasonal Autoregressive Moving Average (SARIMA), Autoregressive Moving

Average with Exogenous inputs (ARMAX) methods. Ref. [43] applied LSTM and GRU methods under different scenarios in which different time steps and different input combinations were used. They used the methods in hybrid form with lasso regression and partial correlation methods. These supplemantary methods were used to extract features for factors affecting the load consumption. Ref. [44] applied LSTM method in their study. First of all, the authors used Density-based Spatial Clustering Applications with Noise (DBSCAN) method for detecting differences between aggregated loads and individual loads for Australian Smart Grid Smart City data set. LSTM method used for predictions by considering three aspects which were load forecasting for individual households,aggregated loads and aggregating forecast vs forecasting aggregate. LSTM was compared with several outstanding machine learning methods, and outperformed the compared methods in general as per the results given.

[45] proposed a hybrid alogirthm that combines Wavelet Transform (WT), feature selection with LSTM for STLF and electricity price forecasting. Comparison of the proposed method was performed with different time series data sets and different machine learning methods. Ref. [46] used sequence2sequence RNNs for STLF. The authors considered different time resolutions with the used data sets for different prediction settings. Ref. [47] presented a hybrid method which combines different machine learning methods, a decomposition method and LSTM. The method comprises of Empirical Mode Decomposition (EMD), LSTM, similar days' selection, extreme gradient boosting-based k-means algorithm for STLF, and they applied the proposed method to a pıblicly available time serises data set. [48] presented a comparative study of smart building data for STLF task. In the study, comparison of LSTM, Random Forest (RF), Extreme Gradient Boosting (XGB), ARIMA and SARIMA methods were performed. In the first part of the study, ARIMA and SARIMA methods used to identify modeling of the stationarity and seasonality issues in data. Later on, deep learning method and machine learning methods were compared. XGB method outperformed the rest of the methods as per the results obtained. [49]

2.2 DNN and Its Variants in STLF

DNNs are advanced type of neural networks in which depth of networks are increased in contrast to shallow neural networks. This fact is also the central theme in development of variants of deep neural networks such as LSTM, GRU, CNN. By adding more layers to the networks or different training of the networks with more layers, it is aimed to get more abstraction from data, and increase the ability of represantation learning of the neural networks.

[50] proposed an integrated method by combining kernel density estimation, quantiled regression and deep neural networks. In the study, interval predictions were performed by combined deep neural networks and quantile regression method. As per the results obtained in the study, the combined approach outperformed gradient boosting and random forest methods. [51] proposed an integrated method which combines auto-encoder network with Support Vector Regression (SVR). The authors performed input construction by feature extraction through using stacked denoising auto-encoders in load data used, then prediction task was performed with SVR method. The proposed method outperformed single SVR, and NN methods. Ref. [52] proposed a deep ensemble learning method which is based on clustering and Lasso method. The proposed method was compared with quantile regression gradient boosting, quantile LSTM, quantile regression forest for prediction tasks of one-hour-ahead and one-day-ahead by utilizing a real-world load data that belong to Irish Commission for Energy Regulation. The data contain half-hourly load records of 800 residents and 400 SMEs customers. [53] used a public available data set in a machine learning repository. They deployed five different methods for the tasks of VSTLF, STLF and MTLF. Two of the deployed methods were Factored Conditional Restricted Boltzman Machine (FCRBM), and Conditional Restricted Boltzman Machine (CRBM) which are some type of deep neural networks, and the rest of the methods were SVR, NNs and RNNs. FCRBM outperformed the rest of the methods used as per the results obtained. [54] used an integrated methodology to perform one-day-ahead and one-week-ahead forecasting tasks with load data of three USA power grids that is publicly available at PJM electricity market. The used methodology consisted of modified mutual information for data preprocessing and feature selection, factored conditional Boltzman machine for forecasting, and newly proposed genetic wind-driven optimization algorithm for fine tuning of the model parameters. Comparison of the methodology was performed with bi-level, mutual information-based artificial neural network method, NN-based accurate and fast converging method, LSTM method to show the efficacy of the methodology. [55] deployed a hybrid method by combining

EMD and Restricted Boltzman Machine (RBM). Comparison of the hybrid method was performed with some papers that used Australian Electricity Operator Market (AEMO) data, and the deployed method performed better than the compared methods as per the results obtained. [56] deployed a hybrid method for STLF. The authors combined a DNN type of stacked RBMs with genetic algorithm to exploit the potential of DNNs'ability of feature extraction in depth. Four concatenated RBMs performed prediction task while genetic algorithm was deployed for optimization of the weights and thresholds of DNNs. [57] used an integrated method which combines EMD, Deep Belief Networks (DBNs) and Local Predictor. EMD made decomposition of data used, DBN trained the weights of stacked RBMs, and Local Predictor constructed neighbor set among forecast samples, then the selected forecast sample were fed into backpropagation algorithm of NN to perform prediction task. The integrated method yielded better results than Least Squares Support Vector Machines (LSSVR), ANN, and DBN with EMD methods as per the results obtained. [58] proposed a new deep neural network method, and deployed it to two public available data sets. Comparison of the proposed method was performed with the methods in the literature that used the same data sets. [24] applied DNNs and RNNs in their study. They performed feature extractions for the factors that affect load consumption characteristics by making analyses in time and frequency domains, and extracted features wer used in the prediction task. They considered different scenarios in their applications, and DNNs and RNNs produced promising results as per the results obtained. [59] used a hybrid method which combines DBN and Copula method. One-year load data of Texas were used for STLF, the hybrid method used was compared with several machine learning methods, and it performed better than NN, SVR, Extreme Learning Machine (ELM) as per the comparison results. [60] proposed a new loss function in their study. Comparisons of CNN, LSTM and MLP by using different metrics were performed with the proposed loss fucntion. The authors utilized two publicly available data sets for the comparisons, and CNN produced better results than the rest of the methods used as a result of the comparisons. [61] proposed a method which is based on ensemble and residual networks. Two public available data sets were used, and comparisons of the proposed method was performed with GRU, LSTM, a proposed basic structure based GRU and LSTM, and several other machine learning methods. As a result of the comparisons, the proposed method outperformed the rest of the methods used. [62] proposed a hybrid method that composes of DBN based EMD and ensembling technique. The authors used AEMO data set to show the applicability of the proposed method by performing comparative study with ELM, RF, SVR, DBN and EMD-based SVR. [63] proposed a novel method called Deep Neural Networks Centered Architecture Evolution (DNN-CAE). The novel method was based on neural architecture search that uses centered architecture evolution for spotting the best DNN architecture for STLF task performed for a real-world case data. The authors made comparison of the proposed method with several different MLP with different sttings, LSTM with different settings, and DNN-CAE shared wegihts method.

2.3 CNN in STLF

CNNs proved their efficiency for many tasks which its efficiency stems from its ability of local feature extraction, then use it for the whole picture by parameter sharing. Extracting underlying temporal correlation and/or spatial correlation in processed data is mainly based on the mentioned ability of CNNs.

Ref. [64] proposed a CNN model based on bagging technique. The authors reshaped load data used by transforming them from one-dimensional data into two-dimensional image data to benefit from correlation between consecutive input sequences. The proposed model deployed inb three phases which the first phase trained CNN model with a big data set of electricty load, the second one included generating subsets of the data used by sampling newly obtained data, and making fine-tuning of CNN bagging models. As a resultant of the second phase applied, sub-models of weak forecasting were created, then these sub-models were merged to create a strong forecasting model. The proposed method produced better results than LR, NN, SVR, and LSTM in the study. Ref. [65] were performed a comparative study by deploying several deep learning methods and machine learning methods for a real-world data set collected in Jiangsu Province in China. They made one-day-ahead prediction by comparing ARIMA, conventional regression models of Multiple Linear Regression (MLR), Regression Tree, Multivariate Adaptive Regression Splines (MARS), SVR, Gaussian Process Regression (GPR), NN, DNN, and a CNN-based model. MLR, SVR and MARS outperformed the rest of the methods used as per the results obtained in the study. [66] proposed a CNN method that uses two-dimensional CNNs for a real-world case application for power consumption data belonging to Algeria. The authors performed VSTLF task by 15-min-ahead forecasting, and STLF task by one-day-ahead forecasting in the study. They made comparisons of the results of the proposed method with some other intelligent methods used in the extant literature.

[67] proposed a new method by combining fuzzy sets with CNN. In the proposed method, fuzzy logic was used for avoiding the over-fitting problem. Hourly electricity load of a power supply system in Malaysia were used, and one-hour-ahead prediction task was performed. The authors compared the proposed method with SARIMA, probabilistic weighted fuzzy time series method, weighted fuzzy time series method, integrated weighted fuzzy time series method, LSTM and some of its variants. [68] proposed a hybrid method that combines CNN with k-means clustering in their study. In the proposed method, k-means clustering was used for grouping data into smaller subsets according to their similarities since the data processed consist millions of data points collected from different regions, and they in turn exhibit different patterns. The load data from a power industry belonging to 2012-2014 with volume of more than 1.4 million data points were used, and comparisons of the method with linear regression, SVR and NN were performed. As a result of the comparisons, the proposed method yielded better results than the rest of the methods used in the study. [69] used a pyramidal architecture of CNN layers for data obtained from SGs for STLF task. Subsets of customers based on their similarities were constructed by utilizing DBSCAN. CNN extracted relevant features from the data for each cluster constructed, then the pyramidal CNN model performed predictions for the clusters for energy-profile customers.

2.4 CNN-RNN Combinations in STLF

RNN, its variants and CNN can be used in hybrid form to take their advantages synergistacally to exploit their natures of efficient sequence modelling, and feature extraction for load forecasting tasks, in turn, they may produce promising results by this application setting.

[70] proposed a hybrid method by blending an attention-based CNN with LSTM and bi-directional LSTM to make short-term load forecasting for an integrated energy system. Feature extraction from data was performed by CNN, LSTM combined with bi-directional LSTM layers made the forecasting. A real-world case application of an integrated energy park in North China was used, and comparisons of the hybrid method with some deep learning methods and machine learning methods were performed.Ref.

[71] compared Seasonal Autoregressive Integrated Moving Average with Exogenous variable (SARIMAX), newly proposed gated RNN and gated CNNin their study. In the comparisons, recursive and direct multi-step forecasting settings were adapted, and the methods were deployed for three real data sets collected in some US schools for the tasks of one-hour-ahead and one-day-ahead predictions. The methods were compared by their accuracies, computational efficacies, generalizations and robustnesses in the study. Ref. [72] proposed time-dependent CNN (TD-CNN) by exploiting the feature extraction ability of CNNs, and Cycle LSTM (C-LSTM) to extract information about time dependence in sequential data, and create shorter number of time steps in their study. One-dimensional load data transformed into two-dimensional images to be used in STLF task, and the authors performed four experiments with the data. The performance comparisons of the proposed method was made with LSTM method in the study. Ref. [73] used a hyrbid form of CNN and RNN. In this hybrid form, parallelly structured CNN modules were used for extraction of valuable features from sequential data while RNN was used to spot and include temporal dynamics of load data used. Weather informations including hourly temperature and humidity, holidays, and special hour(s) or day(s) as inputs for the method. Three-year load data that belong to a city in North China were used in this study, and the proposed method outperformed the compared methods which they are linear regression, SVR, CNN-RNN, DNN. Ref. [74] proposed a hybrid method by combining RNN with one-dimensional CNN, and the authors deployed the method for half-hourly power usage data in South Korea for one-day-ahead forecasting task. Comparisons of the proposed method with MLP, RNN, 1-D CNN were performed, and the proposed method outperformed the rest of the methods compared as per the results obtained. In addition to one-day-ahead forecasting task, 3-day-ahead, 5-day-ahead and 7-day-ahead forecasting tasks were performed in this study. Ref. [75] used one-dimensional data transforming into two-dimensional one. In this study, CNN was used to extract underlying latent features in data, then LSTM performed prediction task with these extracted data. The proposed method was compared with MLP, ResNet, LSTM, and ResNet-MLP for STLF task, and the proposed method outperformed the rest of the methods used in the study as per the results obtained. Ref. [76] compared nine different RNN/CNN and their hybrid forms in their study by utilizing five different data sets, and they presented some concluding remarks about prediction tasks with the methods used in the study. Ref. [77] used hybrid CNN-LSTM

model for STLF of individual households. Firstly, a clustering technique was applied to spot number of outliers, and to discover regularity in daliy power consumotion profiles of individual household data. Next, a hybrid model was proposed which is based on a combination of LSTM and CNN. The developed framework was tested on a publicly available residential smart meter data from Smart Grid Smart City (SGSC) project initiated by the Australian Government. Ref. [78] proposed a hybrid method which combines EMD and similar day methods with CNN-LSTM method. Extracting underlying multimodal spatio-temproal features in used load data was aimed by the proposed hybrid method. Comparative experiments were conducted by using a real-world electricity load data from electricity marker in Singapore. The proposed method was compared with time-dependency CNN, CNN-LSTM, ResNet-LSTM combined model, Similar day based Wavelet Neural Networks (SIWNN), and the proposed method outperformed the compared methods as per the results given.

2.5 Nexus of the Thesis with Related Works

Several different frameworks for the related works are presented in previous sub-sections which shows RNN, LSTM, GRU, CNN, DNN, and their variants' applications for STLF task. The related works use etiher single method or hybrid methods. From the related works, it is obvious that use of deep learning methods for STLF produce promising results when compared with use of conventional methods for STLF, hence deep learning methods supersede conventional machine learning methods in STLF area as well as other areas. By considering this fact, proposing a novel method based on deep learning methods for a real-world case problem is aimed in this study.

Even CNN method uses are extant in the literature, their volumes are smaller when compared to RNN, LSTM, GRU, and their variants' uses. However, CNN use for STLF has gained momentum recenntly, and CNN use for STLF will be likely to surpass the uses of other deep neural networks for STLF. Video Pixel Networks (VPNs) method proposed by Ref. [79] proved its efficacy for video processing for which the method deals with four-dimensional data to generate video frames in simple term. The main part of this method is based on CNNs, and in these CNNs, newly introduced residual multiplicative blocks that contain modified gating mechanisms were utilized. When the extant STLF literature is combed in this thesis scope, VPNs based CNN deployment for STLF task is not available. Hence, we aim to deploy this novel method by some modifications in its gating mechanism for STLF task in this study.

Some distinctive sides of this thesis in relation to the extant litearture are as follows:

- This is the first study which deploys VPNs based one-dimensional CNN for STLF in the literature,
- All deployed methods in this study, mainly deep learning methods, provide an end-to-end learning example unlike hybridized methods. By keeping this genuine, and providing an intiutive method for STLF deployments, this study provides a stepping stone for researchers in STLF area to make contributions to the developed method to extend it, and, develop and compare new methods with the developed method,

In addition to the contributions to the literature, this study provides insight into the decision makers in STLF area by presenting extant off-the-shelf methods of deep neural networks and a novel one.

3. METHODOLOGY AND APPLICATIONS

In this section we introduce our methodolgies for applications. To pursuit this goal, we present data collection, and data pre-processing stages which include feature identification and providing, periodicity and seasonality identifications, framing our data set into a supervised data set are given in detail. Accordingly, methods used in the comparative studies are introduced in this section, and deep learning methods are especially given in detail since the central theme in this study is their applications for an STLF task for a real-world case. In the application stage, model settings for each of the methods are given.

3.1 Data Collection and Pre-processing

In this study, hourly electricity load and temperature data belonging to the period of 2015-2017 collected in Istanbul,Turkey were used as the real-world case problem. We signed a confidentiality and nondisclosure agreement so that the extracted and tabulated data were provided us in a raw format by the electricity provider firm. Since the firm uses the temperature values of the relevant data along with load demand values, the granularity of the data given to us contains the same values as well for our utilization. In this sub-section, we will present data normalization, lag identification, dummy variable inclusion as the data pre-processing tasks [26].

To facilitate numerical stability in computation in the training of neural networks and fast convergence of the methods, we performed feature normalization with a zero-mean normalization procedure for both load and temperature values given in Equation 3.1:

$$x_i' = \frac{x_i - \mu}{\sigma} \tag{3.1}$$

where x_i', μ and σ denote to zero-mean value of i^{th} training example, x_i, in a time series dataset, the mean, and standard deviation of the dataset, respectively. After that, underlying periodicity, cycles, pattern shifts in a time series dataset plays a landmark roles for model construction phases to obtain the desired results as a result of method deploymnets for STLF task. Hence, we examined these in detail by presenting some visuals for them. First of all, we plotted the whole time series dataset which is depicted in Figure 3.1.

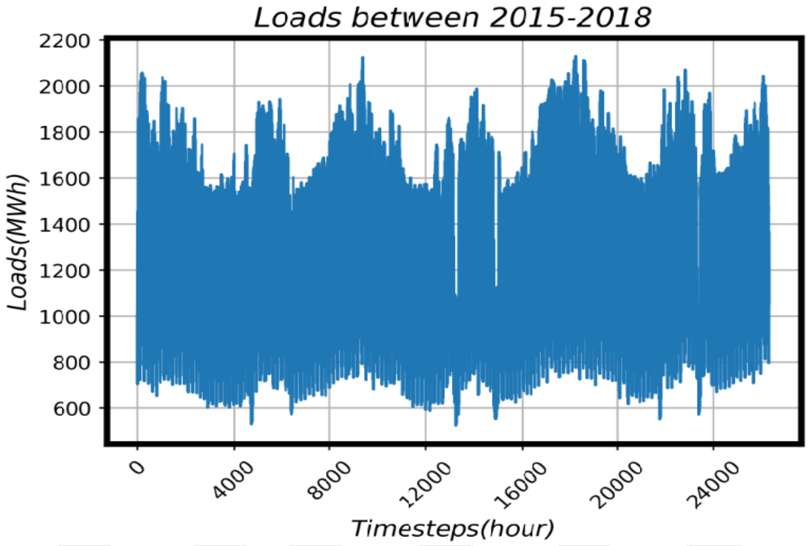

Figure 3.1 : The whole time series.

According to Figure 3.1, the time series has a stationary mean. Seasonal, weekly and daily cycles are available in the series that must be considered in modelling stage. Weekly cycles for a month is provided by Figure 3.2.

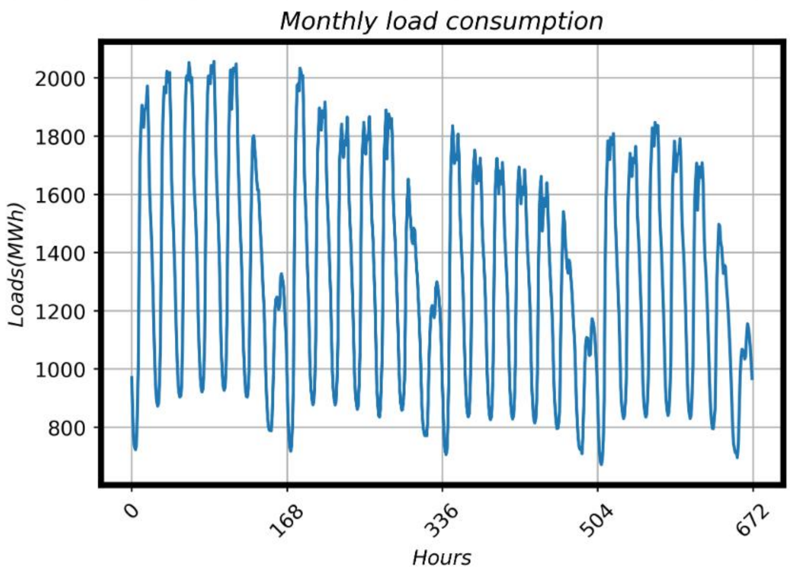

Figure 3.2 : Weekly cycles.

Including seasonality for STLF task is not prevalent since its effects are not so powerful in prediction task, and the firm doesn’t use it in their predictions in real-world as well, however, it is not fully disregarded in the studies in the extant litearture. For a preliminary examination of the methods used in this study, this fact is considered whether the inclusion is worthy for getting better results. As a result of this preliminary examination, the models including seasonality effects did not produce promising result with respect to it non-inclusive counterpart, hence we did not include it modeling stage. On the other hand, daily cycles play another landmark role for STLF task.Thus, we inserted the effects of daily cycles by dummy variables. Providing dummy variables for daily cycles improved the prediction results since distinct consumption characteristics are obvious in the dataset which they are depicted in Figure 3.3.

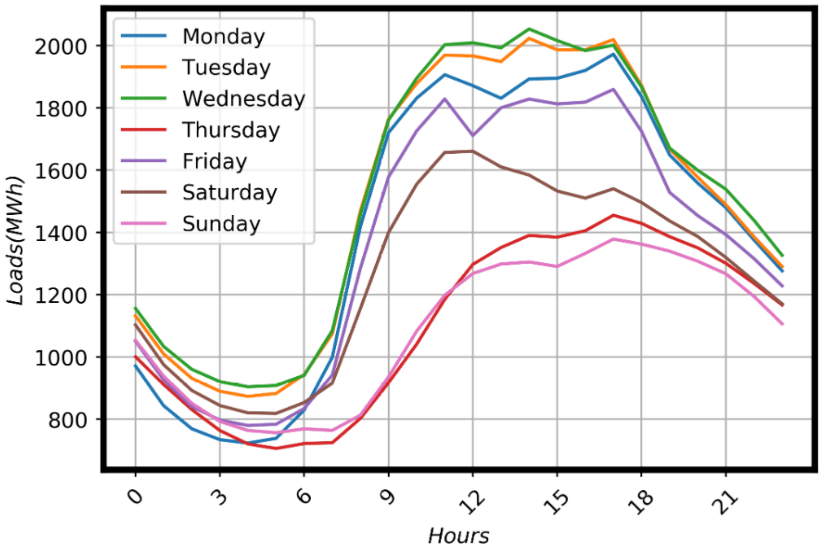

Figure 3.3 : Daily cycles in the series.

Moreover, we investigated lag value identification for the dataset. By examining the lag value for the dataset, we tried to justify the use of 168 time lags for the methods. Hence, we plotted Partial Autocorrelation Function depicted in Figure 3.4.

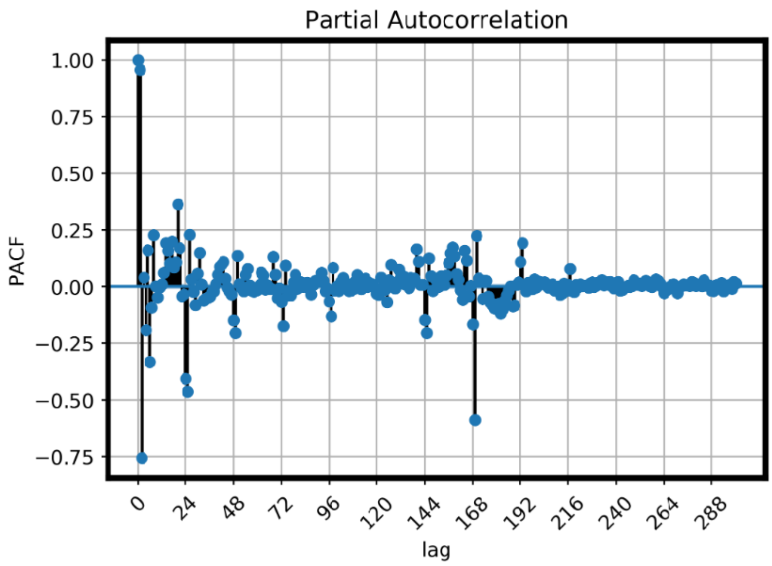

Figure 3.4 : PACF results for the series.

As seen from Figure 3.4, *168* time step back values affect the value to predicted in *169*th time step as a result of Autoregressive (AR) process. PACF results presented in Figure 3.4 justifies the plausibility of previous *168* time steps of the value to be predicted beside the majority of STLF studies provide this same lag value in their method deployments in the literature.

After performing these data pre-processing tasks, we performed converting the time series data set into a supervised data set. Each input feature were shifted by one-time-step-ahead for its relevant output, then the time series data set was in turn framed as a supervised learning problem that is the suitable structure for deep learning methods deployed in the study. In machine learning methods and some of deep learning methods deployed application, that is the first appliaction, the data set was also framed into a supervised learning problem by providing them as a two-dimensional matrix rather than a tensor used with deep learning methods. An examplary input set, and the corresponding output for the inpur set were given in Equation 3.2.

$$Input = \begin{bmatrix} D_{t-168,1} & \cdots & D_{t-168,7} & T_{t-168} & L_{t-168} \\ D_{t-167,1} & & D_{t-167,7} & T_{t-167} & L_{t-167} \\ \vdots & \ddots & & \vdots & \\ D_{t-2,1} & \cdots & D_{t-2,7} & T_{t-2} & L_{t-2} \\ D_{t-1,1} & & D_{t-1,1} & T_{t-1} & L_{t-1} \end{bmatrix}, Output = L_t \qquad (3.2)$$

where dummy variable for a day is represented by *D*, temperature value of a day is represented by *T* while load value of a day is represented by *L*. $D_{i,j}$ denotes to dummy variable for day *j*'s *i*th time step, T_i and $L_i$denote the temperature and the load values of the time step *i*, respectively. Here, L_t denotes the load value to be predicted through using a relevant input set. As a result, constructed input set forms a matrice that is $168x9$ dimensional. For the deployments of the deep learning methods, 1-D CNN based on VPNs use this data set with a tiny difference by providing the demand, load and dummy variables separately in the first layer of the method, then concatantes them to feed this concatenated form into successive layers in it. Hence, the method tries to get more abstraction about the load and the temperature by local feature extraction through treating the input set fed into the first layer of the method in an exogenous variable processing manner. In this variable processing manner, the outputs of load and the temperature were convolved separately, then the resultant outputs of the load and the temperature was concatenated along with raw dummy variables. This idea was also deployed for LSTM, GRU, and their variants deployed in the study to exploit the nature of the exogenous variable providing manner, however, it did not perform better than the original setting that is $168x9$ dimensional matrix providing to the methods.

Prediction strategies are also considered for this study. Since we perform multi-step-ahead prediction task in addition to one-step-ahead prediction task, we have to use one of the prediction strategies for multi-step-ahead predictions. Different prediction strategies have been proposed for the task of multi-step-ahead prediction in the literature [71], [80]. The pravelantly used multi-step-ahead prediction settings are recursive strategy, direct strategy, and Multi-Input Multi-Output (MIMO) strategy in the literature.

Multi-Input Multi-Output (MIMO) Strategy: A given multi-input sequence performs prediction task for a whole output sequence with a single shot by constructing an individual model for the prediction task.

Since we aim to perform one-day-ahead-prediction task along with one-hour-ahead prediction task, the study entails choosing one of the strategies mentioned above for multi-step-ahead prediction task in this study. MIMO strategy does not fit used data, and direct strategy use entails much more complicated model construction for every individual predictions, and entails burdensome computations for each model constructed, thereby increasing computation time significantly in total. Moreover, direct strategy cannot include AR process for the whole sequence to be predicted that a pre-defined time lags affect the value to be predicted. The most suitable strategy for the multi-step-ahead prediction for this study appears as the recursive strategy. Recursive strategy has both computational time advantage, and inherent AR process inclusion for the prediction task. For these reasons, we utilized the recursive strategy for all multi-step-ahead predictions tasks in this study [26].

3.2 Background of the methods used

In this sub-section, we give background information about the methods used. We made two comparative studies throughout this thesis. First one contains deployment of frequently used machine learning methods and some of deep learning methods as preliminary comparative analysis. This deployment was made to show the superiroity of the deep learning methods over conventional machine learning methods. After proving the superiority, we aimed to compare some of the existent deep learning methods with newly introduced 1-D CNN method which is the main theme of this thesis. We partitioned this sub-section into two sections. First section includes background information about machine learning methods deployed, and second section includes background information about deep learning methods deployed. The second section contains more theoretical background while the first one contains the background information straightforwardly with respect to the deep learning methods since the first section methods has supplementary role instead of main one in this thesis. In the first section, SVR, ELM, Gradient Boosting Method (GBM), Extreme Gradient Boosting Method (XGB), Light Gradient Boosting Method (LGBM), RF were presented. Since the central theme in this study is deployment of existing deep learning methods and newly introduced 1-D CNN method, their background information are given in detail with respect to the machine learning methods used in the thesis.

3.2.1 RF, GBM, XGBoost, and LGBM

A model which is made up of many models is called ensemble model. Within the scope of machine learning models, two kinds of ensemble learning are available. One of these is bagging while the other one is boosting. In the boosting, a set of weak learners evolves over time, and forms a model that has strong learners at the end of evolution over time through using weighted average of the learner algorithms iteratively. Any succesive model teachs the upcoming model, then learners are boosted by this teaching mechanism. On the other hand, random sampling with replacement is used to form a learning model in boosting, and this sampling process is repeated many times to form a learning model in boosting. In this mechanism, a sample is chosen randomly from a data set used, then this sample is fitted to a regression or classification tree as per the task of the method used. Randomly choice of a sample with replacement done for boosting is called bootstrapping [81]. Ensemble learning methods are frequnetly used for both of classification and regression tasks, and one of the areas in which ensemble learning methods used is STLF in the literature.

RF method is a kind of bagging method, and classification and regression tasks can be performed by the method. Several samples which are randomly selected from a training data set by bootstrapping which has independent and identically distributed feature are chosen to form a set of prediction trees. Then, Classification and Regression

Tree (CART) algorithm is deployed to this set of prediction trees formed by bootstrapping. By each set of prediction trees, the resultand predictions of the formed trees are collected, and final predictions of these trees are aggregated to get a result of the method.

In RF method, there are a few important points to be considered in modelling. One of them is to identify the best cutting number that is where the node cretaed in a tree will be split. Selection among the features of a model is done to spot the best cutting number. Another important point in RF method is out-of-bag-error (OOBE).OOBE can be considered as the generalization error which is the performance of a method for an unseed data set, and gauges the generalization of RF method. More theoretical information about RF methods can be found in [82].

Gradient Boosting Method (GB), Extreme Gradient Boosting (XGB), Light Gradient Boosting Method (LGBM) are kinds of boosting methods as their name suggests in ensemble learning categories. In boosting, the model's evolves over time, and the final model is formed by constructing a set of stronger learners from a set of weak learners iteratively throughout this evolution process. GBM is based on additive models in boosting, and may perform both of classfication and regression tasks. The additive model constructs GBM works by forward step-wise manner, and loss function is minimized by utilizing selected weak learner, current model and corresponding fit to the model in each step of model construction phase. The solution of the minimization problem is made by gradient descent algorithm which is the prevalently used iterative algorithm in machine learning domain. In RF method, Variable Importance (VI) is an important factor. Splits for each tree to get the loss function minimized are performed by computing the VI for each variable. A variable having high importance contributes to minimization of the loss function more, and makes more improvement of the model predictions while less important variable one makes less contributions to both minimization of the loss function, and improvement of the model predictions. The more VI has a variable, the more VI it will have [83]. More information about GBM can be found in Ref. [84].

Extreme Gradient Boosting (XGBoost) method is another boosting type in ensemble learning categories. The method was proposed by Ref. [85], was based on decision trees. In general, GBM and XGBoost works in the same way, however, XGBoost differs from GBM in a few manners. First difference is that XGBoost uses second order derivative of the loss function unlike GBM. This second order derivative use boosts the performance of XGBoost by enabling it to get into a better places in the loss surface. Second difference is that XGBoost's providing L1 and L2 regularization. This regularization prevents overfitting of model thereby improving the generalization ability of the model. Third difference is that XGBoost provides parallel computing, and this parallelization in computing reduces the burdensome of the computations reqiured by the model especially caused by introduced second order derivation taking. More information about XGBoost method can be found in [85].

Another boosting type of method used in this thesis is LGBM. Computation of information gain of each possible split points for all training examples in a data set creates a computational time complexity thereby this computational time burdensome turns out to be a problem in model deployment for GBM. LGBM addresses this problem by introducing two novel approaches. One of them is Gradient-based One-Side Sampling (GOSS), and the other one is Exclusive Feature Bundling (EFB) [86]. GOSS aims to put emphasis on under-trained training examples with less change in their distributions. Augmentations for the sampled data with small gradients are performed by GOOS approach, and this approach contributes to inclusion of the under-trained examples into model by the augmentations since the largest information gain is considered by making split at each node in a decision tree [86]. On the other hand, EFB approach aims to reduce feature space into a single feature space. This is because mutually exclusive feature space creates a sparse feature space, and hence bundling exclusive features into an individual feature will speed up computation by decreasing the computational complexity from $O(\#data\ x\ \#features)$ into the one $O(\#data\ x\ \#bundle)$. In this complexity measures, the number of bundles are pretty much less than the number of features. These mentioned approaches contributes to efficacy and scalability of the high feature spaces, reducing the computational burdensome thereby emerging as an alternative method for machine learning applications. More information about LGBM can be found in Ref. [86].

3.2.2 SVR

Support Vector Machine (SVM) method was proposed by Ref. [87]. This method is based on statistical learning theroy, and used in many works for both classification and regression tasks even the method was originally developed in the context of classification tasks. By proving its efficacy in classification tasks, the method was extended for regression tasks, and dubbed Support Vector Regression (SVR). The motivation behind SVR is to construct a robust regression which reduces the effects of outliers in the regression task dealt with [88]. SVR provides a unique solution for a problem since it delas with a constrained optimization problem if the properties of the problem are met. The basic idea in SVR deployment is that it tries to solve optimization problem by a non-linear mapping of an input set into a higher dimensional feature space by utilizing kernel trick. It also benefits from duality theorem of the optimization theory in its mechanism [89]. More information about SVM and SVR can be found in [84].

3.2.3 ELM

[90] proposed a distilled version of neural network by introducing ELM. A new training alogirthm for a single-hidden layer feed-forward neural networks was proposed by the authors. In this training scheme, hidden nodes in a single-hidden layer feed-forward neural networks are chosen randomly, then analytical solutions for outputs of the networks are performed unlike the regular training algorithms does it iteratively. By this manner, the problem is converted into a leas squares problem, hence an analytical solution is obtained for the networks. ELM can be used for both classification and regression tasks.

In most of problems that use the neural networks, number of training examples are more than number of hidden nodes. Hence, the matrix of hidden nodes to be inversed will not be a square matrix thereby reaching the zero training error under this condition is impossible. ELM helps the network be an over determined linear system for solution, and it provides a special solution by utilizing Moore-Penrose (MP) which is a generalized inverse used for the hidden nodes in the neural networks by ELM.

3.2.4 LSTM and GRU

Long-term computing dependency in sequential data makes to process the data and get promising results hard for shallow neural networks. RNNs provides a solution for this type of data by introducing recurrence of feedback within layers of the neural networks by a time delay ability [38]. In a typical RNN, output value at time step t is computed by means of input value at current time step and output value conveyed from earlier time steps obtained by the recurrences. A hidden state computation at time step t for an RNN is given in Equation 3.3.

$$h_t = f(w_i x_t + w_h h_{t-1} + b) \tag{3.3}$$

where h_t is the hidden state value at time step t, x_t and h_{t-1} denote the input value at current time step t and the output value (hidden state) of one time step before, respectively. w_i and w_h denote weights for the input, and the hidden state of the previous time step, respectively. b denotes bias term for a hidden layer, and f denotes a non-linear activation function.

As it was mentioned that RNNs are superior over feed-forward neural network in processing sequential data that have long term dependency, even RNN may suffer from this long term dependency. This long term dependency creates gradient vanishing and exploding in backpropagation since too small numbers or too large numbers will be multiplied within a chain of sequence in the succesive layers. Hence, gradients will explode or vanish. As a result, RNNs efficacy will degrade. Within these problems, much more complicated one is vanishing gradient problem since “exploding gradient problem” was addressed by a simple solution that is gradient clipping proposed by [91]. On the other hand, attempts to address “vanishing gradient problem” have been made by researchers. In these attempts, use of gating mechanisms to prevent the problem is the central theme. LSTM proposed by [92] and GRU proposed by [93] addressed “vanishing gradient problem” by utilizing gating mechanism in RNNs, and got promising results. LSTM method has some variants in which some modifications are done by some merging operations in its gating mechanisms. Clockwork RNN, Depth-Gated LSTM, and GRU is some of the modified versions of LSTM, however, the most prevalently used modified version of LSTM is GRU method. Hence, we will present LSTM and GRU methods, which are the subject of this thesis, in detail.

An LSTM cell is comprised of input, output and forget gates. Each of these gates some part of information of the LSTM cell. An exemplary visual of an LSTM cell is depicted in Figure 3.5.

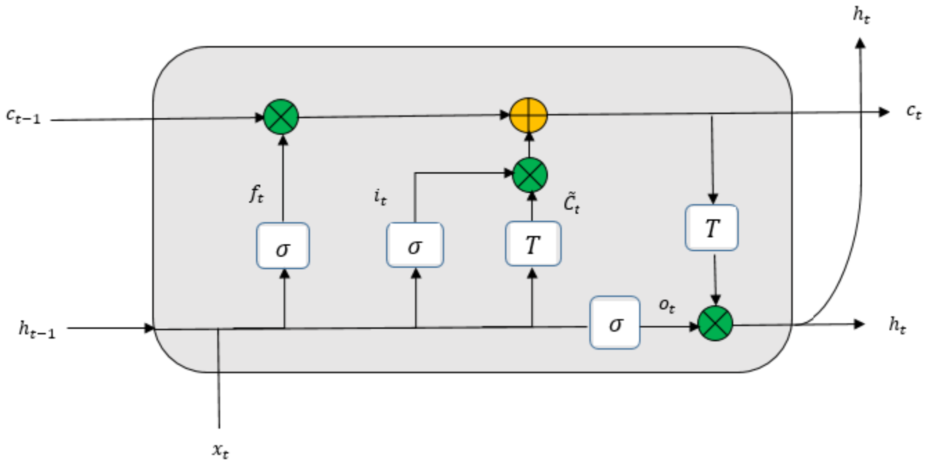

Figure 3.5 : An LSTM cell.

According to Figure 3.5, input state and hidden state are concatenated where they are denoted by x_t and h_{t-1} in Figure 3.5, respectively. Accordingly, this concatenated inputs is taken in a sigmoid function, hence forget gate which spots the information to be retained and forgotten is formed. This gate is denoted by f_t in Figure 3.5, and its computation is given in Equation 3.4.

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{3.4}$$

In Equation 3.4, W_f and b_f denote weights and bias term for the forget gate, respectively. σ denotes sigmoid activation function, and concetanation operation is symbolized by the square bracket.

Cell state denoted by c_t is rsponsible for storing the relevant information to be carried into successor layers of LSTM. Cell state is formed by f_t, c_{t-1}, i_t and $\tilde{C}_t$. $c_{t-1}$and $\tilde{C}_t$ correspond to previous time step's cell state and candidate values conveyed for the next cell state, respectively while i_t corresponds to input gate. Computations for an input gate and a candidate values are given in Equations 3.5-3.6.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3.5}$$

$$\tilde{C}_t = T(W_c[h_{t-1}, x_t] + b_c) \tag{3.6}$$

In Equations 3.5-3.6, weights for input gate and matrices of candidate values are denoted by W_i and W_c, respectively. Bias terms of input gate and matrices of candidate values are denoted by b_i and b_c, respectively. T corresponds to hyperbolic tangent activation function. Cell state computation is given in Equation 3.7 in which $\odot$ represents element-wise multiplication.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{C}_t \tag{3.7}$$

In Equations 3.8-3.9, output gate and hidden state for current time step computations are given, respectively.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.8}$$

$$h_t = o_t \odot T(c_t) \tag{3.9}$$

GRU is a modified version of LSTM, and it merges forget and inputs gates into a single update gate z_t. In addition to that, cell state and hidden state are also merged in GRU. GRU is comprised of two gates which are update and reset gates while LSTM has three gates. In GRU mechanism, what information to be conveyed through the succesive time steps are spotted by update gate. On the other hand, what information to be forgotten are spotted by reset gate, r_t. An exemplary visual of an GRU cell is depicted in Figure 3.6.

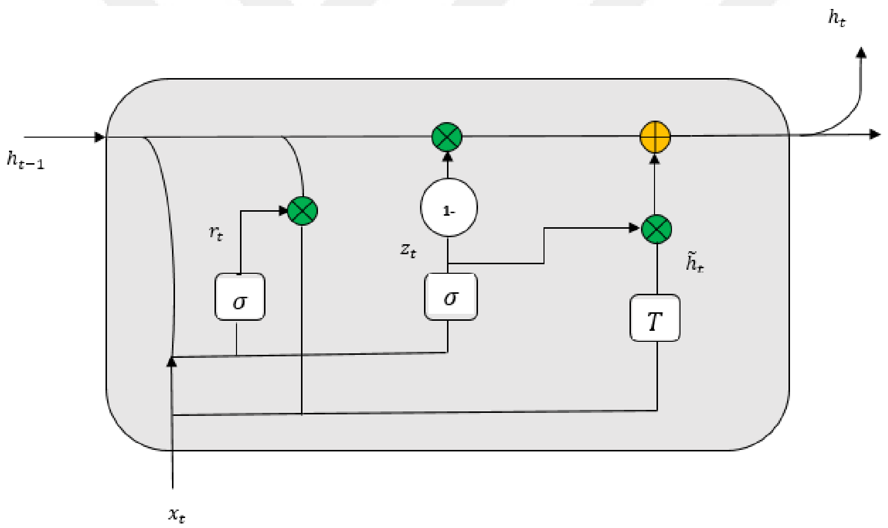

Figure 3.6 : A GRU cell.

In the following equations, Equations 3.10-3.11 give computations for update and reset gates in order, and Equations 3.12-3.13 give computations for current memory content and final memory content, respectively.

$$z_t = \sigma(W_z[h_{t-1}, x_t]) \tag{3.10}$$

$$r_t = \sigma(W_r[h_{t-1}, x_t]) \tag{3.11}$$

$$\tilde{h}_t = T(W[r_t * h_{t-1}, x_t]) \tag{3.12}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{3.13}$$

In this thesis, stacked LSTM and stacked GRU methods with three layer stacks, and their encoder-decoder versions are deployed. Stacked LSTM and encoder-decoder LSTM methods are depicted in Figure 3.7 and Figure 3.8, respectively.

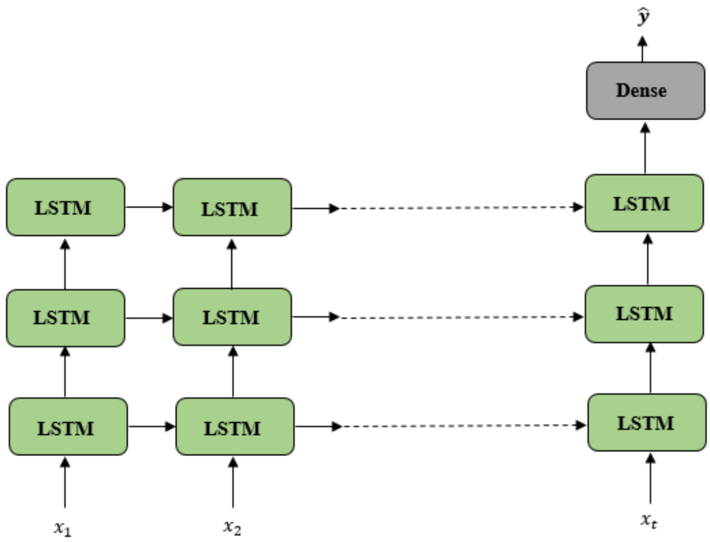

Figure 3.7 : Stacked LSTM.

In the interest of brevity, stacked GRU visual is not given since replacement of LSTM cells with GRU cells in Figure 3.7 gives the visual for stacked GRU.

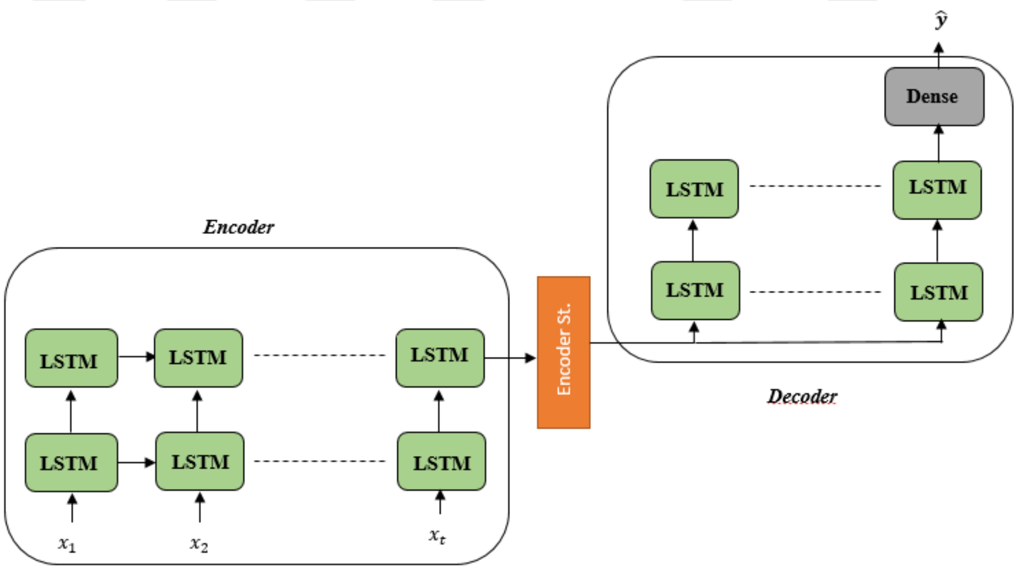

Figure 3.8 : Encoder-decoder LSTM.

In the interest of brevity again, visual for encoder-decoder GRU is not given since replacement of LSTM cells with GRU cells in Figure 3.8 gives the visual of encoder-decoder GRU. In Figure 3.7, each of rectangle boxes corresponds to an LSTM cell depicted in Figure 3.5. Here x_t denotes input of time step t. Dense box at the top of the rightmost of the figure refers to a fully connected layer, and $\hat{y}$ corresponds to predicted value by the method. In Figure 3.8, encoder-decoder model is comprised of two parts. The encoder state part includes stacked parts of cells, and encoder state is constructed by the last LSTM cell's output computed at the end of the sequence of the first stacked part in the encoder. Accordingly, this encoder will feed the decoder part which is a stacked part as well, and predictions will be performed by this part. Since the information stored in the encoder part feeds the decoder part, and contributes to its representation learning ability, the encoder-decoder architecture has proven its efficay in representation learning, and it produces promising results alongside stacked versions of LSTM and GRU. Thus, we include the encoder-decoder architecture as well as stacked architecure. Since we will use lag value as 168 time steps, x_t will correspond to 168^{th} input instance of an input set. In other words, x_t in Figures 3.7-3.8 will correspond to 168^{th} input instance in stacked LSTM, stacked GRU, encoder-decoder LSTM, and encoder-decoder GRU.

3.2.5 Proposed 1-D CNN based on VPNs

CNNs are different kind of neural networks which benefits from local connectivity between layers by shared parameters. CNNs are used in many areas such as computer vision, image recognition, pattern recognition, healthcare, robotics, natural language processing. CNN deployments seems to be rare when compared to LSTM and GRU deployments for time series problems, however, its deployments for time series problems have gained momentum in recent times in the literature [68]–[70], [77], [94], [95]. By considering this momentum combined with CNNs' power, and the main theme of this thesis, we aim to propose a novel method of CNN for a real-world case of STLF, and compare the proposed method's with with exsiting deep learning methods ,and show the applicability and efficacy of it. As mentioned in earlier section of the thesis, we try to develop 1-D CNN method based on VPNs which is a powerful methods for video processing to address time series forcasting problem. In this sub-section, background of the proposed method will be given in detail.

A CNN typically applies convolution operation with a fixed length kernel, k, by sliding over an input x with a cretain stride. Hence, by this overlapping of each kernel on the input patches, a representative output of the input is generated as a result of this convolution operation. In this operation, element-wise multiplication of kernel with the input patches is performed, then the resultant outputs of this multiplication is summed up, hence feature maps of the input are cretaed which is mentioned as the representative output. The depth of output is relied on number of filters used in the convolution operation. On the other hand, the spatial dimensions of the feature maps are relied on stride for the convolution and padding for the input [26].

1-D CNN used in the proposed method is regarded as the suitable one for time series type of problems. The convolution operation with a one-dimesional CNN is given in Equation 3.14.

$$g_i = (x * w)(i) = \sum_{k=0}^{f-1} x(i-k)w(k) \tag{3.14}$$

where x corresponds to one-dimensional input that is $x \in R^d$, w corresponds to weight for the one-dimensional kernel applied that is $w \in R^f$ for the convolution's i^{th}element.

One type of convolution is causal convolutions which they do not violate the order of data processed thereby not allowing a value to be predicted reliance on values of future time steps [96]. An illsutrative example of a causal convolution structure is given in Figure 3.9.

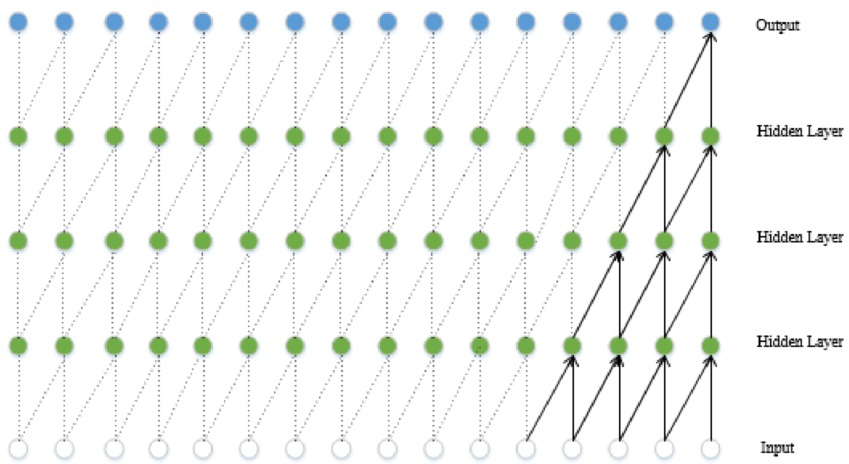

Figure 3.9 : Structure of causal convolutional layers.

One advantage of using causal convolutional layers instead of RNNs is that they do not contain reccurrent conncetions within their layers thereby their compuation times are significantly lesser than the ones of RNNs. When long sequences are modeled, superiority of the causal convolutional layers over RNNs appears saliently. However, there arises the risk of augmenting the receptive fields of a network as a result of entailment of using many layers and/or large filters in causal convolutions. To address this problem, dilated convolutions comes in for solution in CNNs. It has similar manner with causal convolutional layers except for the fact that the convolution operations are performed over an input by striding with a determined stride rate [96]. An illustrative example of structure of a dilated causal convolutional layers is depicted in Figure 3.10.

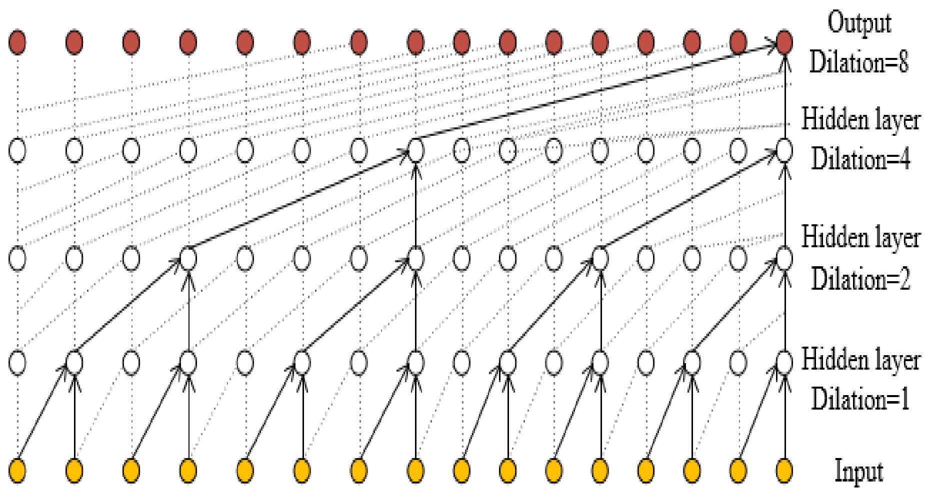

Figure 3.10 : Structure of a dilated causal convolutional layers.

A dilated causal convolution operation for an input x with weight w and dilation rate d is given in Equation 3.15.

$$h_i = (x *_d w)(i) = \sum_{k=0}^{f-1} x(i - d_k)w(k) \tag{3.15}$$

where the input $x \in R^d$, the weight for one-dimensional kernel is $w \in R^f$. Receptive fields grows exponentially in dilated causal convolutions while their growth is linear for causal convolutions. Exponential growth rate is computed as $r = 2^{L-1}k$, and the linear one is computed as $r = k(L-1)$ where L corresponds to the length of the sequnce (time-series data length) processed. Long-range temporal dependencies available in sequences is efficiently processed by this exponentially grown receptive fields. This growth also helps the network preserve resolution of input, work with fewer parametrs thereby contributing the computational efficiency of the network [96]. Besides the dilated convolutions, residual connections are also another remarkable property for CNNs. Depth is important feature in deep neural networks since making abstractions in low/mid/high levels primarily depends on depth of a network. Hence, it may boost performance of the network. Recently, architectures of networks that has many depths have been producing promising results through using aforementioned residual connections [97], [98]. However, deepening the networks may cause degradation of these deep networks thereby making training of the deep networks hard. This degradation problem is created by adding more layers to the networks rather than being as a result of conventional overfitting problem of neural networks. This problem in turn affects training error of the networks. [97] proposed an approach to address this problem by introducing identity mapping of added layers. Desired underlying mapping of an input H_x is defined as $F(x) \coloneqq H(x) - x$, and original mapping is recasted into $F(x) + x$. This recasted mapping facilitates identity mapping within succesive layers of the deep networks [97],[26]. Hence, providing these identity mappings into the neural networks prevents degradation problems in the neural networks, and contributes to performance of the networks. An illustrative example of such an identity block is depicted in Figure 3.11.

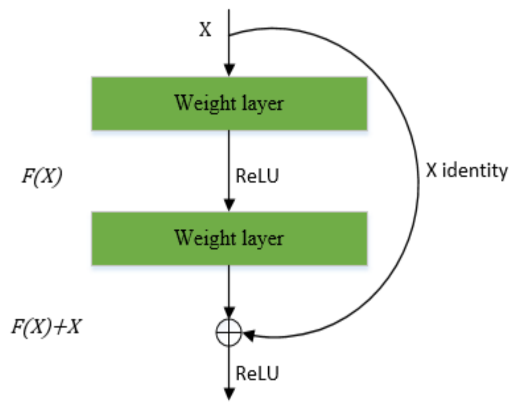

Figure 3.11 : An identity mapping by a residual connection.

Residual Multiplicative Block (RMB) and Multiplicative Unit (MU) modules proposed by [79] are another main ingredients used in the proposed 1-D CNN method. They were originally developed for video processing task, however, we adopted MU and RMB ideas with some modifications in their gtaing mechanisms for the proposed method. Rather than deploying VPNs proposed by Ref. [79] directly, we deployed a variant of this method for the essence of proposing a novel method in this thesis. A typical MU has gates merged into a convolutional layer. This convolutional layer takes in an input h which has $T\ x\ c$ size. Here T corresponds to length of the input given, and c corresponds to size of channels in the convolutional layer. A MU takes in the input, and four gates are structured by this input which are update gate denoted by u, and other gates g_{1-3}. Consecutively, output of the MU is produced by element-wise multiplications and summations performed on these gates [79], [26]. Computations for the gates and resultant output are given in Equations 3.15-3.19.

$$g_1 = \sigma(W_1 * h) \tag{3.15}$$

$$g_2 = \sigma(W_2 * h) \tag{3.16}$$

$$g_3 = \sigma(W_3 * h) \tag{3.17}$$

$$u = ReLU(W_4 * h) \tag{3.18}$$

$$MU(h; W) = g_1 \odot ReLU(g_2 * h + g_3 \odot u) \tag{3.19}$$

where $\odot$ corresponds to element-wise multiplication, and $ReLU$ denotes rectified linear unit activation function.

In the proposed 1-D CNN method, we replaced $tanh$ activation function in the original paper with $ReLU$ presented in Equations 3.18-3.19. When tanh activtion is used in update gate and for the resultant output of a MU, vanishing gradient may be encountered for deeper layers thereby affecting the performance of the neural networks. Hence, we proposed to replace tanh activations there with $ReLU$ with the aim of introducing more robust gradients againist vanishing problem where $ReLU$ does not attenuate magnitude of gradients of the input unlike tanh activation , and we in turn got better results than the original setting of VPNs [26].

One of the main ingredients in the proposed methods is RMB module. In this module, consecutive two layers are stacked. After this stacking, gradient propogation throughout the networks used is performed easily by means of a residual connection between input and output in this module. At the beginning part of a MU layer, channel sizes of an input h is halved. This operation is done by performing $1x1$ convolution with linear transformation. Subsequently, resultant output of the first MU is conveyed to second MU layers for processing. At the end of the second MU layer, $1x1$ convolution with a linear transformation is done. As a result of this transformation, the size of channels of resultant output of the sesond MU are doubled to comply with the input input h. Finally, the resultant output and the input are added by using residual connection [79].

Computations for an RMB module are given in Equations 3.20-3.24.

$$h_1 = W_1 * h \tag{3.20}$$

$$h_2 = MU(h_1; W_2) \tag{3.21}$$

$$h_3 = MU(h_2; W_3) \tag{3.22}$$

$$h_4 = W_4 * h_3 \tag{3.23}$$

$$RMB(h; W) = h + h_4 \tag{3.24}$$

A MU structure is depicted in Figure 3.12.

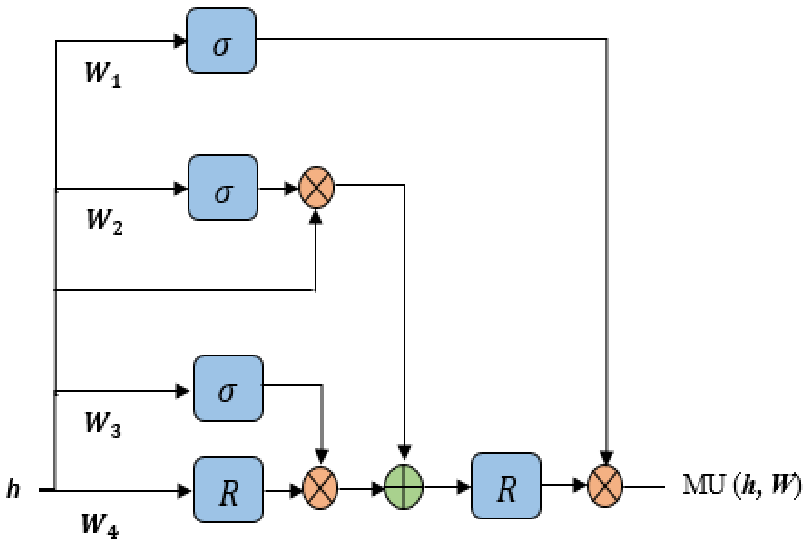

Figure 3.12 : Structure of a MU layer.

In Figure 3.12, R denotes $ReLU$ activation function, and σ denotes sigmoid activation function.

An RMB structure is shown in Figure 3.13.

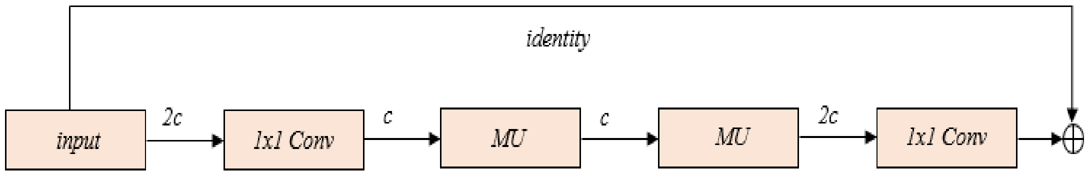

Figure 3.13 : Structure of an RMB.

As mentioned before, an RMB module is comprised of two succesive MU layers which have one convolution layers before and after them that controls the channel size of the input provided to MU layers. By merging residual connection idea, dilation and causal onvolutions along with MU layers and RMB modules, we generate the proposed method which its architecture is depicted in Figure 3.14 [26].

In Figure 3.14, L denotes load input for a provided training dataset while T and H denote temperature value, and hour cycle value for the training dataset. On the other hand, D denotes dummy variables for weekdays. FC denotes a fully connected layer for a network, and $\hat{y}$ denotes predicted value by the method. We set the method to to process inputs in exogenous variable manner to get more abstraction from data in earlier layers. Hence, load data is provided seperately in the first layer of the proposed method, and temperature data along with hour cycles are provided into other branch of the method as seen from Figure 3.14. After this processing, extracted feature maps of the data for each feature are concetaned with dummy variables for weekdays, and this whole input is fed into subsequent layers of the network [26].

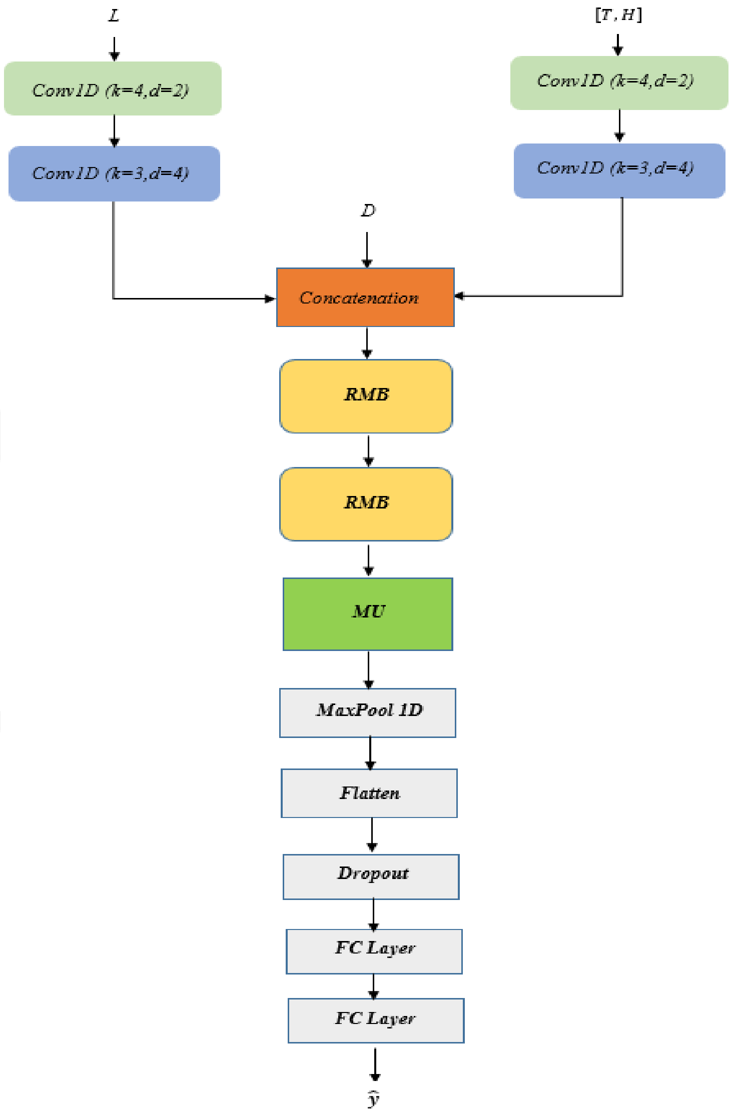

Figure 3.14 : The proposed 1-D CNN method architecture.

3.3 Applications and Comparative Studies

In this sub-section, we provide implementation details for the comparative studies, and discuss their results in depth. Throughout all computations for all the methods deployed in this thesis, we utilized Keras 2.2.2 with Tensorflow 1.10.0 as backend in Pyhton 3.6 enviroment.

In the first comparative study, we compared stacked LSTM, stacked GRU, ELM, SVR, RF, GBM, XGB, LGBM. In all studies, we used Mean Squared Error (MSE) metric for loss functions, it penalizes the error quadratically theeby performing efficiently with respect to its conuterprts. Hence, we selected it for the loss functions. On the other hand, Mean Absolute Percentage Error (MAPE) metric was used for comparing results of the methods deployed in each comparative study since it is commnly used metric by the provider firms. MSE and MAPE computations are given in Equations 3.25-3.26, respectively.

$$\mathrm{MSE} = \frac{1}{K}\sum_{i=1}^{K}(y_i - \hat{y}_i)^2 \tag{3.25}$$

$$\mathrm{MAPE} = \left(\frac{1}{K}\sum_{i=1}^{K}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) * 100 \tag{3.26}$$

where K denotes size of the data set, y_i and $\hat{y}_i$ correspond to actual value and predicted value of the time step i in the data set, respectively.

For each of the methods deployed, extensive parameter search by the grid search for each method was conducted, and their best settings were included studies. We run each of the methods ten times to get an ensemble results for them since randomization of parameters of the methods may have huge effects on a single run. Thus, we aim to prevent randomization effect on the method deployments by making ensembling of the individual prediction results. In addition, we present statistical tests for the ensemble results to identify whether the result of the best performing method significantly better than the rest of the methods deployed. For this purpose, MannWhitneyU test for the ensemble results were conducted since this non-parametric test does not entail strict assumptions unlike its parametric counterparts. By taking the advantage of this non-parametric test, we tried to identify the significance of the best performing method with respect to the rest of the methods deployed in the study. In the comparative studies, we present the monthly prediction results for the methods in comparison tables. In the tables, the result of the best performing method was given in bold, and the second one was italicized for convenience of tracking the results. In comparison tables, difference column wass presented in which the percentage of performance difference for the first and the second best performing methods were computed [26].

In Table 3.1, one-hour-ahead predictions of the preliminary comparative analysis was given.

Table 3.1 : One-hour-ahead predictions for the first comparisons.

	GRU	LSTM	LGBM	XGB	GBR	SVR	RF	ELM	Difference (%)
Jan.	*0.94*	**0.93**	1.47	2.78	1.75	2.41	1.94	3.57	1.06
Feb.	**1.00**	*1.03*	1.27	2.96	1.84	1.59	2.23	3.08	2.91
Mar.	*1.06*	**1.04**	1.09	2.70	1.70	1.29	1.91	3.04	1.89
Apr.	**0.93**	*0.98*	1.19	2.92	1.87	1.33	2.53	3.29	5.1
May	*1.07*	**1.03**	1.50	3.43	2.17	1.64	2.57	3.70	3.74
Jun.	**1.09**	*1.10*	2.21	3.67	2.82	2.41	3.20	4.60	0.9
Jul.	**0.95**	0.99	1.64	3.47	2.16	1.79	2.50	3.48	4.04
Aug.	**0.96**	*1.01*	1.26	3.32	2.04	1.24	2.47	2.88	4.95
Sep.	*1.10*	**1.01**	2.03	4.01	2.51	2.68	2.99	4.32	8.18
Oct.	*0.88*	1.03	**0.87**	2.57	1.31	1.00	1.72	2.99	1.14
Nov.	*0.98*	**0.85**	1.06	2.51	1.56	1.05	1.76	2.80	13.27
Dec.	*1.04*	**1.00**	1.24	2.91	1.78	1.49	2.07	3.13	3.85
Mean	**1.00**	**1.00**	*1.40*	3.10	1.96	1.66	2.32	3.41	28.57

Stacked GRU and stacked LSTM methods outperformed the rest of the methods as per the results given in Table 3.1. Only month that the overall best performing methods falls behind the best performing one is October. However, the difference is minuscule according to difference column. On the other hand, the best performing methods majorly made predictions around 1% while the others' predictions were dispersed. Mainly, the performance differences between the best performing one and its succesor method are clear for the months of April, August, September, November. The overall difference between the best performing methods, stacked GRU and stacked LSTM, and LGBM is 28.57%. The performance differences between the methods in September and August are noteworthy. A religious holiday and a national holiday concatenated when the latter one fell on the end days of August, and the former one fell on the first days of September. Their combined effects have hugely impacted the prediction ability and performance differences of the methods. The difference on November is between the top two methods, it may have been caused by a random effect since there is no clear explanation for difference.

A graphical illustration for the one-hour-ahead predictions is depicted in Figure 3.15.

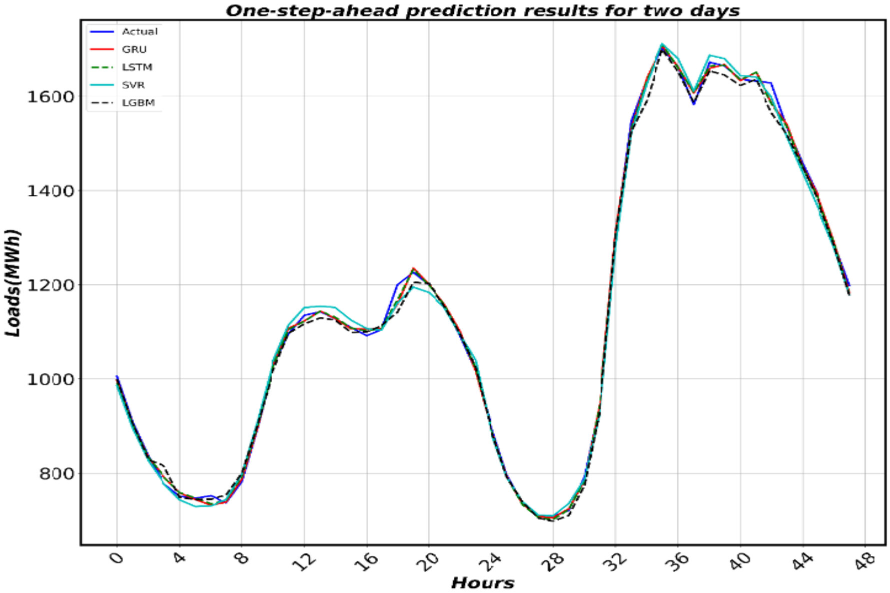

Figure 3.15 : One-hour-ahead predictions for the first comparisons.

Top four performing methods are depicted in Figure 3.15. After predictions, significance of the performance differences will be investigated by MannWhitneyU test. Results for the tests are given in Table 3.2.

Table 3.2 : First test results for the first comparisons.

Methods	p-value	Methods	p-value
S.GRU-S.LSTM	0.155	S.LSTM-S.GRU	0.155
S.GRU-LGBM	0.00017	S.LSTM-LGBM	0.00018
S.GRU-XGB	0.00016	S.LSTM-XGB	0.00018
S.GRU-GBR	0.00017	S.LSTM-GBR	0.00018
S.GRU-SVR	0.00005	S.LSTM-SVR	0.00006
S.GRU-RF	0.00016	S.LSTM-RF	0.00017
S.GRU-ELM	0.00017	S.LSTM-ELM	0.00018

In the tests, p-value as equal to 0.01 is used to keep confidence inetrval tight. As per the results given in Table 3.2, there is no an obvious performance difference between stacked GRU and stacked LSTM, they both perform well for one-hour-ahead predictions. On the other hand, they outperformed the rest of the methods for this prediction task as the p-values are siginificantly smaller than the confidence interval. Hence, by regarding the results in Table 3.2, we conclude that deep learning methods outperformed the conventional machine learning approaches in one-step-ahead prediction task with the data. As the second deployment for the first section of the comparisons, we present multi-step-ahead predictions of deep learning methods and machine learning methods. Table 3.3 presents one-day-ahead prediction results.

Table 3.3 : One-day-ahead predictions for the first comparisons.

	GRU	LSTM	LGBM	XGB	GBR	SVR	RF	ELM	Difference (%)
Jan.	**2.65**	3.06	*2.69*	5.12	2.96	3.49	4.36	4.47	*1.49*
Feb.	**2.06**	2.28	*2.21*	5.63	3.38	2.43	4.71	3.97	*6.79*
Mar.	*2.07*	2.14	**1.87**	4.68	2.79	2.26	3.72	4.03	*9.66*
Apr.	**1.94**	*2.01*	2.21	4.81	3.15	2.37	4.11	4.39	*3.48*
May	**2.63**	*2.74*	3.30	6.05	3.97	3.60	4.54	5.51	*4.01*
Jun.	5.36	**4.33**	*5.06*	6.87	5.42	5.76	5.97	7.67	*14.43*
Jul.	**2.92**	*2.94*	4.24	6.84	5.19	3.42	6.21	4.80	*6.8*
Aug.	*2.82*	3.02	2.94	6.42	4.36	**2.66**	5.82	3.96	*5.67*
Sep.	*4.80*	**4.62**	5.62	8.81	6.71	5.38	9.08	6.45	*3.75*
Oct.	**1.18**	*1.30*	1.35	3.59	1.96	2.12	2.64	3.90	*9.23*
Nov.	**1.36**	*1.72*	1.88	3.88	2.56	1.91	3.20	3.66	*20.9*
Dec.	**1.66**	*2.01*	2.29	4.96	2.85	2.86	4.23	4.12	*17.41*
Mean	**2.62**	*2.68*	2.97	5.64	3.78	3.18	4.88	4.74	*2.24*

Results in Table 3.3 is somehow different from the results in Table 3.2 with respect to some months. This may be because multi-step-ahead prediction is more challenging than one-step-ahead predictions. Reguarly, multi-step-ahead predictions produce more error than one-step-ahead predictions. As per the results given in Table 3.3, the most successive method emerges as stacked GRU with a total MAPE of 2.62. It is followed by stacked LSTM with a total MAPE of 2.68. The difference between the best performing method and its successor is 2.24 % in total. Unlike one-step-ahead prediction results, performance difference of predictions for September and August is relatively low with respect to high error having months in the predictions. However, prediction results are very high for September when investigated in methodwise manner. The best performing method's MAPE result is 4.62, and this result may be caused by the aformentioned holiday concatenations at the end of August and at the beginning of September. The other noteworthy prediction error appears on June. This high error rate may be attributable to summer season since load consumption pattern distincly can change. In addition, some part of June contained the Ramadan month that night consumption patterns change. This effect may be combined with summer season's effect, and they may have synergystical effects on the load predictions. This rare time issue modelling in load prediction area especially dual calendar effects remains an open area for researchers.

We present a visual for one-day-ahead predictions for the first comparions by the best performing 4 methods in analyses in Figure 3.17.

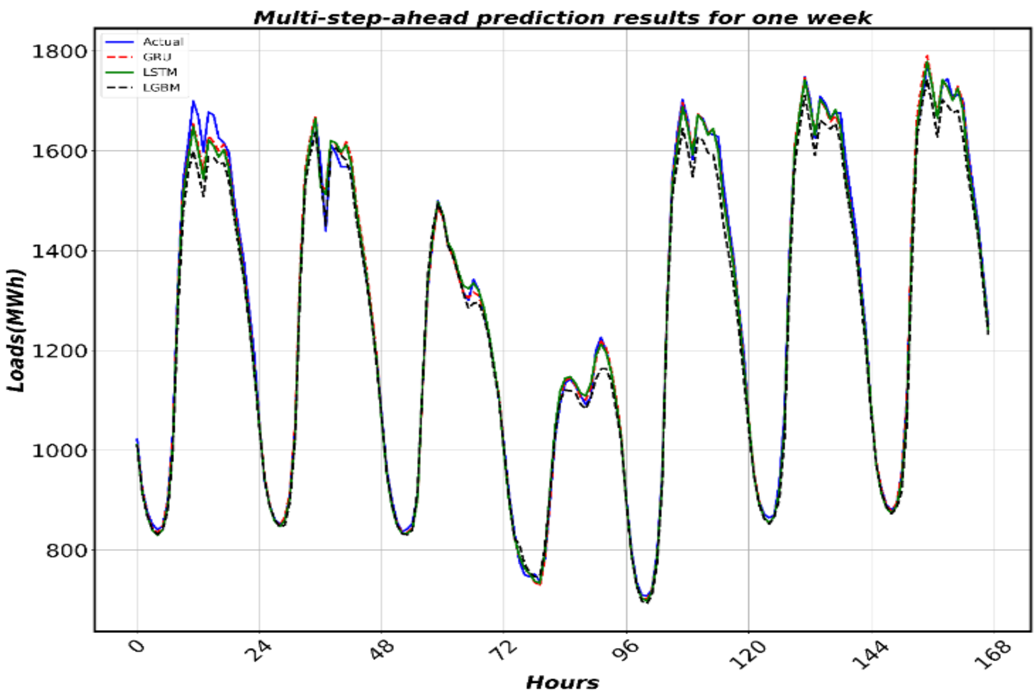

Figure 3.16 : One-day-ahead predictions for the first comparisons.

We repeated the test with the same p-value whether the best performing method's results significantly differ from the rest of the methods. The test results are presented in Table 3.4.

Table 3.4 : Second test results for the first comparisons.

Methods	p-value
S.GRU-S.LSTM	0.342
S.GRU-LGBM	0.0018
S.GRU-XGB	0.00013
S.GRU-GBR	0.00013
S.GRU-SVR	0.00005
S.GRU-RF	0.00014
S.GRU-ELM	0.00015

As per the results obtained in Table 3.4, the ensemble performance of stacked GRU is significantly better than the rest of the methods except for stacked LSTM. We have no evidence to reject the null hypothesis that the results of stacked LSTM and stacked GRU significantly differ from each other. As a result of the first comparsions, deep learning methods significantly outperforms the conventional machine learning methods for STLF with the data used. Hence, we extend our study to compare deep learning methods. In these comparisons, stacked GRU, stacked LSTM, encoder-decoder GRU, encoder-decoder LSTM, and the proposed 1-D CNN method will perofrm the tasks for STLF predictions.

In Table 3.5, we present one-hour-ahead prediction results for deep learning methods.

Table 3.5 : One-hour-ahead predictions for the second comparisons.

	S.GRU	S.LSTM	E.D.GRU	E.D.LSTM	1-D CNN
Jan.	0.99997	0.99997	0.99997	0.99997	0.99997
Feb.	1.00002	1.00002	1.00002	1.00002	1.00002
Mar.	1.00012	1.00012	1.00012	1.00012	1.00012
Apr.	1.00033	1.00033	1.00033	1.00033	1.00033
May	1.00054	1.00054	1.00054	1.00054	1.00053
Jun.	1.00055	1.00055	1.00055	1.00054	1.00054
Jul.	1.00022	1.00022	1.00022	1.00022	1.00022
Aug.	1.00017	1.00017	1.00017	1.00017	1.00017
Sep.	1.00044	1.00043	1.00044	1.00043	1.00043
Oct.	1.00037	1.00037	1.00037	1.00037	1.00037
Nov.	1.00013	1.00013	1.00012	1.00012	1.00013
Dec.	1.00003	1.00003	1.00003	1.00003	1.00003
Mean	1.000241	1.000240	1.000240	1.000238	1.000238

As per the results obtained in Table 3.5, prediction performance of the methods are nearly the same for one-step-ahead prediction tasks since they are powerful methods for this task, and their feature extraction abilities far beyond the conventional machine learning methods. The results of the first comparisons showed this fact clearly in the previous analyses. Table 3.6 presents the test results of one-hour-ahead predictions for the second (main) comparisons.

Table 3.6 : First test results for the main comparisons.

Methods	p-value
1-D CNN-S.GRU	0.368
1-D CNN-S.LSTM	0.942
1-D CNN-E.D.GRU	0.942
1-D CNN-E.D.LSTM	0.190

Since the prediction error differences are minuscule for all the methods, we directly give the test results without presenting difference column for the method comparisons in Table 3.5, and presented test results in Table 3.6. As per the results presented in Table 3.6, there is no significance difference between methods in the task of one-step-ahead prediction. A visual for this one-step-ahead predictions is depicted in Figure 3.17.

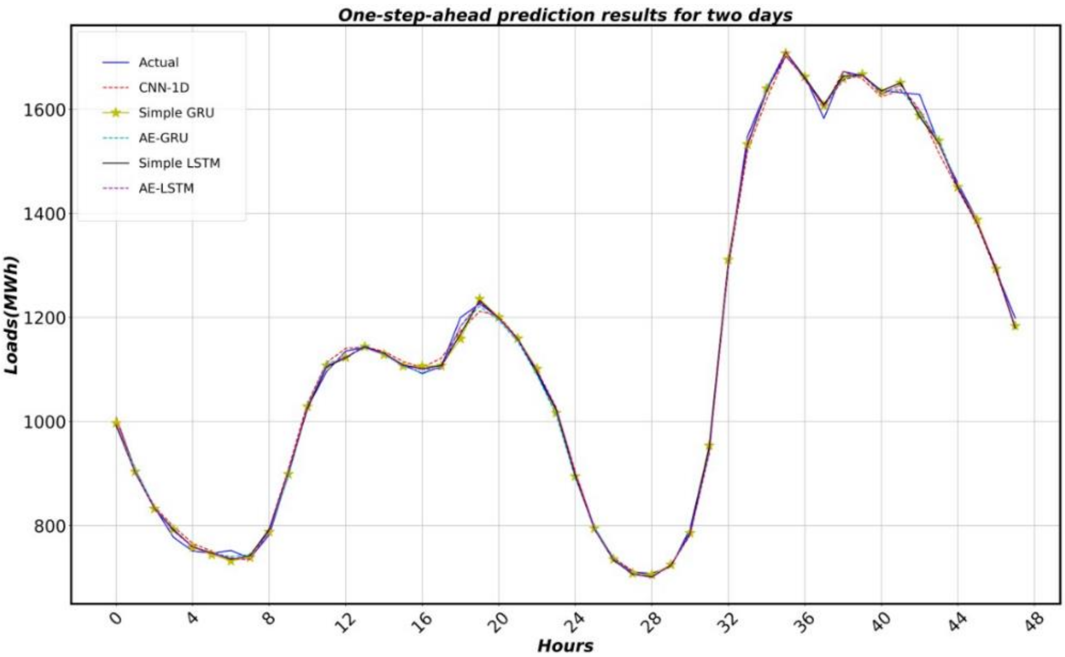

Figure 3.17 : One-hour-ahead predictions for the second comparisons.

One-day-ahead prediction results for the second comparisons are given in Table 3.7.

Table 3.7 : Second prediction results for the main comparisons.

	S.GRU	S.LSTM	E.D.GRU	E.D.LSTM	1-D CNN	Difference (%)
Jan.	*2.65*	3.06	3.39	3.07	***2.43***	***8.3***
Feb.	*2.06*	2.28	2.35	2.16	***1.74***	***15.5***
Mar.	2.07	2.14	*2.05*	2.06	***1.56***	***23.9***
Apr.	1.94	2.01	1.96	*1.87*	***1.54***	***17.6***
May	*2.63*	2.74	3.02	2.68	***2.55***	***3***
Jun.	5.36	4.33	4.68	*4.12*	***3.95***	***4.1***
Jul.	2.92	2.94	3.56	*2.70*	***2.47***	***8.5***
Aug.	2.82	3.02	3.09	*2.62*	***2.25***	***14.1***
Sep.	4.80	4.62	*4.50*	4.66	***3.88***	***13.8***
Oct.	1.18	1.30	1.18	*1.13*	***1.10***	***2.7***
Nov.	*1.36*	1.72	1.49	1.63	***1.24***	***8.8***
Dec.	***1.66***	2.01	1.88	1.89	*1.79*	*7.3*
Mean	2.62	2.68	2.76	*2.55*	***2.21***	***13.3***

Unlike Table 3.5 results, results in Table 3.7 show the noteworthy differences between the best performing method, the proposed 1-D CNN, and the rest of the methods for the task of one-day-ahead predictions. As per the results presented in Table 3.7, the proposed method outperforms the rest of the methods for all months except for December. Stacked GRU outperforms the method in this month. In total, the performance difference of the proposed method with respect to its successor is 13.3% as given in the last row of the difference column in Table 3.7. Predictions for June and

September are high as well as multi-step-ahead predictions of the first comparisons. The same situation is encountered here as it will be mentioned in conclusions. The test results for multi-step-ahead predictions for the main comparisons are given in Table 3.8.

Table 3.8 : Second test results for the main comparisons.

Methods	p-value
1-D CNN-S.GRU	*0.00018*
1-D CNN-S.LSTM	*0.00018*
1-D CNN-E.D.GRU	*0.00025*
1-D CNN-E.D.LSTM	*0.00033*

A visual for multi-step-ahead predictions is given in Figure 3.18.

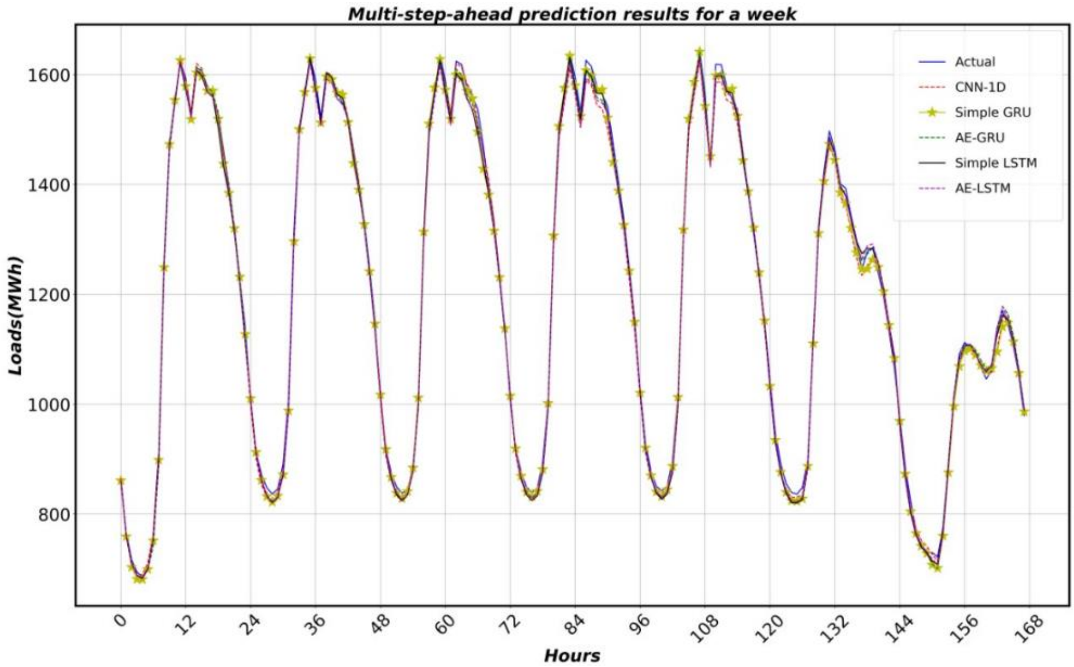

Figure 3.18 : One-day-ahead predictions for the second comparisons.

As per the results given in Table 3.8, the proposed method has significantly better results than the rest of the methods. The method outperformed the rest of the methods in one-step-ahead predictions, however, performance differences were not clear, and there was no significant difference between the methods. In contrast, the method dominantly outperforms the rest of the methods according to the ensemble results for the multi-step-ahead predictions, and the test results justify efficacy of the proposed method for this prediction task. In the following sub-section discussions will be given in brief.

3.4 Dicussions of the Results

Two comparisons are made for this thesis. First one which is the supplementray one to the second (main) comparisons contains the comparisons deep learning and machine learning methods for the tasks of one-step-ahead prediction and multi-step ahead prediction. According to the results of the one-step-ahead prediction and multi-step-ahead predictions for the fitst comparisons, deep learning method, stacked GRU, outperforms the rest of the methods.In these first comparisons, the performance differences between the best performing method, deep learning, and the conventional machine learning methods are spotted significant. One exception is that stacked GRU outperformed stacked LSTM as well, but the performance differences between two deep learning methods are not spotted as statistically significant in these tasks. As a result of the first comparisons, two deep learning methods proved their efficacies. Then, we extended the scope of the deployment to the main comparisons. In these main comparisons, the proposed 1-D CNN method slightly goes beyond the other methods in the prediction results, and performance differences of the predictions between the proposed method and the rest of the deep learning methods are so minsucule. Hence spotting any performance differences between deep learning methods deployed does not produce any desired statistically significant result for one-step-ahead prediction task. On the other hand, according to the multi-step-ahead predictions deployed as one-day-ahead predictions, the proposed 1-D CNN method outperformed the rest of the methods, and its prediction performance difference suggests that there are significantly difference between the results of the proposed method and the rest of the methods. At the end, the proposed 1-D CNN method produced better results than the extant deep learning methods for the application data. As per the results of the main comparisons, proposed 1-D CNN method emerges as a challenger for extant deep learning methods. In the method-level regarding, all the methods for each comparison, especially in the multi-step-ahead predictions, capturing the rare time issue pattern emerged as a problem for all the methods. Prediction errors for June and September are high with respect to other months. As mentioned before, coincidence of some days of June with Ramadan month may have synergystically affected the load consumption pattern on June with temperature effect. Another noteworthy high errors are available for July, August and September. July and August errors may be relatively low with respect to the errors for September, and these errors may be primarily caused by combined effects of high temeprature and humidity. In the summer seasons, the combined effects lead to extensive air-conditioner use, the load consumption characteristic in turn changes and diverges from the regular pattern. The models may not be spotted these changes and divergences. Another landmark issue in the prediction errors is the September errors. This is caused by two holidays' coincidence on this month. This appears as the second rare time issue caused by dual calendar effects [26].

In the prediction tasks, CNN's feature extraction ability contributes to the proposed method's performances in the comparisons. Even LSTM and GRU have efficiency in capturing patterns in sequential data. CNN also introduces pattern capturing like LSTM and GRU, but local connectivity by parameter sharing and correlation capturing with kernels boost its performance in many tasks including time series forecasting. Hence, the proposed method exploited these natures for CNN, and outperformed the other methods for the data used.

4. CONCLUSIONS

In recent times, the emergence of Artificial Intelligence has been becoming more apparent. Surge in big data regime, novel arcihtecture developments for neural networks, advents in hardware and software supercharge the spread of AI deployments in many industrial areas. In these developments and improvements, neural networks has central importance since deep neuraal networks achieves good performances as compared to their counterparts, machine learning and statistical learning methods, in the context of regression and classification tasks. Supervised, unsupervised and reinforcement learning types are supported by deep neural networks, they in turn get promising results in computer vision, robotics, self-driving cars, natural language processing, audio recognition, speech recognition, and so forth. Furthermore, other industrial areas are also the subject of deployments AI techniques. Short-term electricty load forecasting area is one of the areas that a supervised type of deep learning methods are deployed that is the main aim of this thesis.

In this thesis, first of all, data preprocessing tasks were performed, and feature extraction and preparations for each method were made. These preliminary tasks were followed by first round of method comparisons. In this sense, we compared two deep learning methods, stacked GRU and stacked LSTM, with conventional machine learning methods widely used for STLF whether deep learning outperforms the methods. As a result, the deep learning methods outperformed saliently the compared methods for the tasks of one-hour-ahead and one-day-ahead predictions for which the results were boosted by the statistical significance tests. This first stage is followed by the second round of comparisons. In this sense, deep learning methods were compared. Stacked LSTM, stacked GRU, encoder-decoder LSTM, encoder-decoder GRU, and the proposed CNN 1-D method were compared. As a result, the proposed method outperformed the rest of the methods in one-day-ahead prediction task which is proved by statistical significance test. On the other hand, the proposed method outperformed the rest of the method, however, the performance gain was minuscule to conclude that the proposed method is also superior to the rest of the methods in one-hour-ahead prediction task. This fact is validated by the significance test as well.

This study has versatile contributions for both researchers and decision makers. In this study, an end-to-end deep learning method was proposed for a real-world case, and its efficacy was gauged by comparing it with some other end-to-end deep learning methods. Hence, the results of the study presents a new method for the field of time series prediction. On the other hand, the methods deployed in this study does not contain hybridized methods that may move the practitioners out of the scope of deep neural networks so that deployment of the methods are intiutive for managerial decision makers in STLF area. Furthermore, elaboretaly designed application stages provide insights and guide into the practitioners in this area [26].

Method-level and organizational level takeaways from this study are present as a result of the method deployments. Method-level takeaways may be stated as followings [26]:

- Calendar effects are significant for modelling time series problemsi hence boosts the performance of the methods when properly inserted for the methods deployed,

- With a nexus to above effects, dual calendar issue and its effects have central importance in time series modeling. Hence, this problem can be addressed by different approaches such as some business rules and adjustments,

- Beside recurrent neural network architectures, ability of CNNs appears as a result of this study for time series problems. Not only with time series problems, but also with all sequence problems, CNNs will likely compete with recurrent neural network architectures,

- In this study, CNNs' abilities of spatio-temporal feature extraction available in data and capturing the correlation in long sequences emerge CNNs as a challenger for its cıounterpart deep learning methods,

- The proposed method provides exogenous variable inclusion to modelling the time series problem thereby enhancing the feature extraction ability in early layers of the method by boosting the peroformance of it. This is because using larger time-frequency resolution captures high-level features in advance separately for each feature. This enhancement works for only the proposed method since exgoneous variable inclusion input setting for other deep learning methods did not boost the performances of them.

As the organizational-level takeaways, the followings can be stated;

- Due to big data regime, advents in hardware, GPUs, TPUs, MPPs, and software solutions, Keras, Tensorflow, PyTorch, and so forth, for deep neural networks, and introduction of novel architectures for deep neural networks, creating business value by utilizing the aforementioned advancements will provide many benefits for businesses. Electricty provider firms may be one of the businesses which can benefit from these advanets to create business value in the stated manner by using more intelligent methods such as deep learning and deep reinforcement learning for their operations instead of conventional statistical-based and machine-learning-based methods,

- According to the results obtained in this study, deep learning methods outperformed conventional machine learning methods for STLF. This fact states that using the intelligent methods will contribute to decision making process for predictions in the electricity provider firm. Furthermore, using these methods will contribute different levels of the organizations. Some of the contributions of the accurate predictions are reduction in maintenance costs, stable and secure power supply system, low penalty costs for overestimating or underestimating the load consumption, enhancement in competetiveness, the market share growth.

- Smart Grids, Distributed Generations, Smart Grid Plus, Energy 5.0, the Energy Internet, the IoT concepts in the energy sector also entails intelligent method use for efficient system management with low cost and agile response. Introducing deep learning and deep reinforcement learning methods with available big data for these concepts will enhance competitiveness of the provider firms as well as load prediction tasks,

- This study provides an off-the-shelf method(s) for both tasks of one-hour-ahead and one-day-ahead predictions which are the major concerns for the providers operating in Turkish electricty market regulated by EPİAŞ. Provided off-the-shelf methods can readily be used by decision makers of the provider firms in STLF area in addition to the researchers in this area since it does not require any theoretical knowledge of the methods.

Some future directions can be addressed as a result of this study. Some of these future directions may be as followings;

- To make the method deployment in large scale, real-case data of different regions of Turkey can be used to compare the performance of the methods. Fırthermore, new architectures can be introduced to the comparisons,
- We compared the proposed method with some of the prevalent machine learning methods for STLF. Other machine learning methods can be included for the comparisons such as Gaussian Process Regression, Lasso Regression along with other real-case data,
- Very Short-Term Load Forecasting and Medium-Term Load Forecasting tasks can be preformed by using proper input settings with the proposed method. Furthermore, performance of the proposed method can be compared with other deep learning methods and machine learning methods for these tasks.

REFERENCES

[1] Yang, W., Dong, Y.,Du, Q.,Qiang, Y., Wu, K.,Zhao, J.,Yang, X.,Zia, M.B. (2020). “Integrate domain knowledge in training multi-task cascade deep learning model for benign–malignant thyroid nodule classification on ultrasound images,” *Eng. Appl. Artif. Intell., 98*, 104064, doi: https://doi.org/10.1016/j.engappai.104064.

[2] S. Pang, J. J. del Coz, Z. Yu, O. Luaces, and J. Díez, (2017). “Deep learning to frame objects for visual target tracking,” *Eng. Appl. Artif. Intell., 65*, 406–420.

[3] Dong, Y. (2019). “Implementing Deep Learning for comprehensive aircraft icing and actuator/sensor fault detection/identification,” *Eng. Appl. Artif. Intell., 83*, 28–44.

[4] Gil-Martín, M., San-Segundo, R., Fernández-Martínez, F., Ferreiros-López, J., (2020). “Improving physical activity recognition using a new deep learning architecture and post-processing techniques,” *Eng. Appl. Artif. Intell., 92*, 103679.

[5] Worsham, J. and Kalita, J. (2020). “Multi-task learning for natural language processing in the 2020s: Where are we going?,” *Pattern Recognit. Lett., 136*, 120–126.

[6] Tai, L., Paolo, G., Liu, M. (2017). “Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 31–36.

[7] Chen, J., Yuan, B., Tomizuka, M. (2019). “Model-free deep reinforcement learning for urban autonomous driving,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2765–2771.

[8] Zhao, J., Qiu, G., Guan, Z., Zhao, W., He, X. (2018). “Deep reinforcement learning for sponsored search real-time bidding,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1021–1030.

[9] Hu,Y.-J., Lin, S.-J. (2019). “Deep reinforcement learning for optimizing finance portfolio management,” in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 14–20.

[10] Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N. J., Xie, X., Li, Z. (2018). “DRN: A deep reinforcement learning framework for news recommendation,” in *Proceedings of the 2018 World Wide Web Conference*, 167–176.

[11] Berk, K., Hoffmann, A., Müller, A. (2018). “Probabilistic forecasting of industrial electricity load with regime switching behavior,” *Int. J.*

Forecast., *34* (2), 147–162.

[12] Cancelo, J.R., Espasa, A., Grafe, R. (2008). "Forecasting the electricity load from one day to one week ahead for the Spanish system operator," *Int. J. Forecast.*, *24* (4), 588–602.

[13] Soares, L.J., Medeiros, M.C. (2008). "Modeling and forecasting short-term electricity load: A comparison of methods with an application to Brazilian data," *Int. J. Forecast.*, *24* (4), 630–644.

[14] Hu, R., Wen, S.,Zeng, Z., Huang, T. (2017). "A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm," *Neurocomputing*, *221*, 24–31.

[15] Kavousi-Fard, A., Samet, H., Marzbani, F. (2014). "A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting," *Expert Syst. Appl.*, *41* (13), 6047–6056.

[16] Osman, Z.H., Awad, M.L., Mahmoud, T.K. (2009). "Neural network based approach for short-term load forecasting," in *2009 IEEE/PES Power Systems Conference and Exposition*, 1–8.

[17] Ceperic, E., Ceperic, V., Baric, A. (2013). "A strategy for short-term load forecasting by support vector regression machines," *IEEE Trans. Power Syst.*, *28* (4), 4356–4364.

[18] Hooshmand, R.-A., Amooshahi, H., Parastegari, M. (2013). "A hybrid intelligent algorithm based short-term load forecasting approach," *Int. J. Electr. Power Energy Syst.*, *45* (1), 313–324.

[19] Hewamalage, H., Bergmeir, C., Bandara, K. (2020). "Recurrent neural networks for time series forecasting: Current status and future directions," *Int. J. Forecast*.

[20] Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T. (2020). "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecast.*, *36* (3), 1181–1191.

[21] Cheng, L. and Yu, T. (2019). "A new generation of AI: A review and perspective on machine learning technologies applied to smart energy and electric power systems," *Int. J. Energy Res.*, *43 (*6), 1928–1973.

[22] Ozcanli, A.K., Yaprakdal, F., Baysal, M. (2020). "Deep learning methods and applications for electrical power systems: A comprehensive review," *Int. J. Energy Res*.

[23] Guan, C., Luh, P. B., Michel, L. D., Wang, Y., Friedland, P. B. (2013). "Very short-term load forecasting: wavelet neural networks with data pre-filtering," *IEEE Trans. Power Syst.*, *28* (1), 30–41.

[24] Din, G. M. U. and Marnerides, A. K. (2017). "Short term power load forecasting using deep neural networks," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, 594–598.

[25] Chen,Y., Luh, B., Guan, C., Zhao, Y., Michel, L.D., Coolbeth, M.A., Friedland, P.B., Rourke, S.J. (2009) "Short-term load forecasting: Similar day-based wavelet neural networks," *IEEE Trans. Power Syst.*, *25* (1), 322–330.

[26] Yazici, I, Beyca, O. F., Delen, D. (2022). "Deep-learning-based short-term electricity load forecasting: A real case application," *Eng. Appl. Artif. Intell.*, *109*, 104645.

[27] Djukanovic, M., Ruzic, S., Babic, B., Sobajic, D. J., Pao, Y. H. (1995). "A neural-net based short term load forecasting using moving window procedure," *Int. J. Electr. Power Energy Syst.*, *17* (6), 391–397.

[28] Santos, P. J., Martins, A. G., Pires, A. J. (2007). "Designing the input vector to ANN-based models for short-term load forecast in electricity distribution systems," *Int. J. Electr. Power Energy Syst.*, *29* (4), 338–347.

[29] Sudheer, G. and Suseelatha, A. (2015). "Short term load forecasting using wavelet transform combined with Holt–Winters and weighted nearest neighbor models," *Int. J. Electr. Power Energy Syst.*, *64*, 340–346.

[30] Wang, S., Wang, X., Wang, S., Wang, D. (2019). "Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting," *Int. J. Electr. Power Energy Syst.*, *109*, 470–479.

[31] Mishra, M., Nayak, J., Naik, B., Abraham, A. (2020). "Deep learning in electrical utility industry: A comprehensive review of a decade of research," *Eng. Appl. Artif. Intell.*, *96*, 104000.

[32] Guyot, D., Giraud, F., Simon, F., Corgier, D., Marvillet, C., Tremeac, B. (2019) "Overview of the use of artificial neural networks for energy-related applications in the building sector," *Int. J. Energy Res.*, *43* (13), 6680–6720.

[33] Lee, D., Huang, H., Lee, W., Liu, Y. (2020) "Artificial intelligence implementation framework development for building energy saving," *Int. J. Energy Res.*

[34] Fan, C., Wang, J., Gang, W., Li, S. (2019). "Assessment of deep recurrent neural network-based strategies for short-term building energy predictions," *Appl. Energy*, *236*, 700–710.

[35] Shi, H., Xu, M., Ma, Q., Zhang, C., Li, R., Li, F. (2017). "A whole system assessment of novel deep learning approach on short-term load forecasting," *Energy Procedia*, *142*, 2791–2796.

[36] Fekri, M. N., Patel, H., Grolinger, K., Sharma, V. (2021). "Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network," *Appl. Energy*, *282*, 116177.

[37] Kumar, S., Hussain, L., Banarjee, S., Reza, M. (2018). "Energy Load Forecasting using Deep Learning Approach-LSTM and GRU in Spark Cluster," in *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, 1–4.

[38] Wen, L., Zhou, K., Yang, S. (2020). "Load demand forecasting of residential buildings using a deep learning model," *Electr. Power Syst. Res.*, *179*, 106073.

[39] Narayan, A. and Hipel, K. W. (2017). "Long short term memory networks for short-term electric load forecasting," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2573–2578.

[40] Du, S., Li, T., Yang, Y., Horng, S.-J. (2020). "Multivariate Time Series Forecasting via Attention-based Encoder-Decoder Framework," *Neurocomputing*.

[41] Xiuyun, G., Ying, W., Yang, G., Chengzhi, S., Wen, X., Yimiao, Y. (2018). "Short-term Load Forecasting Model of GRU Network Based on Deep Learning Framework," in *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, 1–4.

[42] Muzaffar, S. and Afshari, A. (2019). "Short-term load forecasts using LSTM networks," *Energy Procedia, 158*, 2922–2927.

[43] Zheng, J., Chen, X., Yu, K., Gan, L., Wang, Y., Wang, K. (2018). "Short-term Power Load Forecasting of Residential Community Based on GRU Neural Network," in *2018 International Conference on Power System Technology (POWERCON)*, 4862–4868.

[44] Kong, W. , Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., Zhang, Y. (2017). "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*.

[45] Memarzadeh, G. and Keynia, F. (2021). "Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm," *Electrical Power Systems Research, 192*, 106995.

[46] Skomski, E., Lee, J.-Y., Kim, W., Chandan, V., Katipamula, S., Hutchinson, B. (2020) "Sequence-to-sequence neural networks for short-term electrical load forecasting in commercial office buildings," *Energy Build., 226*, 110350.

[47] Zheng, J., Xu, C., Zhang, Z., Li, X. (2017). "Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network," in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, 1–6.

[48] Hadri, S., Naitmalek, Y., Najib, M., Bakhouya, M., Fakhri, Y., Elaroussi, M. (2019). "A Comparative Study of Predictive Approaches for Load Forecasting in Smart Buildings," *Procedia Comput. Sci., 160*, 173–180.

[49] Ferdoush, Z., Mahmud, B.N., Chakrabarty, A., Uddin, J. (2021). "A short-term hybrid forecasting model for time series electrical-load data using random forest and bidirectional long short-term memory.," *Int. J. Electr. Comput. Eng., 11*, 1.

[50] Guo, Z., Zhou, K., Zhang, X., Yang, S. (2018) "A deep learning model for short-term power load and probability density forecasting," *Energy, 160*, 1186–1200.

[51] Tong, C., Li, J., Lang, C., Kong, F., Niu, J., Rodrigues, J. J. P. C. (2018). "An efficient deep model for day-ahead electricity load forecasting with stacked denoising auto-encoders," *J. Parallel Distrib. Comput., 117*, 267–273.

[52] Yang, Y., Hong, W., Li, S. (2019). "Deep ensemble learning based probabilistic load forecasting in smart grids," *Energy*, *189*, 116324.

[53] Mocanu, E., Nguyen, P. H., Gibescu, M., Kling, W. L. (2016). "Deep learning for estimating building energy consumption," *Sustain. Energy, Grids Networks*, *6*, 91–99.

[54] Hafeez, G., Alimgeer, K. S., Khan, I. (2020). "Electric load forecasting based on deep learning and optimized by heuristic algorithm in smart grid," *Appl. Energy*, *269*, 114915.

[55] Qiu, X., Ren, Y., Suganthan, P. N., Amaratunga, G. A. J. (2017). "Empirical mode decomposition based ensemble deep learning for load demand time series forecasting," *Appl. Soft Comput.*, *54*, 246–255.

[56] Hui, X., Qun, W., Yao, L., Yingbin, Z., Lei, S., Zhisheng, Z. (2017). "Short-term load forecasting model based on deep neural network," in *2017 2nd International Conference on Power and Renewable Energy (ICPRE)*, 589–591.

[57] Yu, Y., Ji, T. Y., Li, M. S., Wu, Q. H. (2018). "Short-term Load Forecasting Using Deep Belief Network with Empirical Mode Decomposition and Local Predictor," in *2018 IEEE Power & Energy Society General Meeting (PESGM)*, 1–5.

[58] Chen, K., Chen, K., Wang, Q., He, Z., Hu, J., He, J. (2018). "Short-term load forecasting with deep residual networks," *IEEE Trans. Smart Grid*.

[59] He, Y., Deng, J., Li, H. (2017). "Short-term power load forecasting with deep belief network and copula models," in *2017 9th International conference on intelligent human-machine systems and cybernetics (IHMSC)*, *1*, 191–194.

[60] Li, N., Wang, L., Li, X., Zhu, Q. (2020). "An effective deep learning neural network model for short-term load forecasting," *Concurr. Comput. Pract. Exp.*, *32* (7), e5595.

[61] Xu, Q., Yang, X., Huang, X. (2020). "Ensemble Residual Networks for Short-Term Load Forecasting," *IEEE Access*, *8*, 64750–64759.

[62] Fan, C., Ding, C., Zheng, J., Xiao, L., Ai, Z. (2020). "Empirical Mode Decomposition based Multi-objective Deep Belief Network for short-term power load forecasting," *Neurocomputing*, *388*, 110–123.

[63] Gomez-Rosero, S., Capretz, M. A. M., Hydro, L. (2020). "Deep neural network for load forecasting centred on architecture evolution," 19th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE.

[64] Dong, X. , Qian, L., Huang, L. (2017). "A CNN based bagging learning approach to short-term load forecasting in smart grid," in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 1–6.

[65] Liu, X., Zhang, Z., Song, Z. (2020). "A comparative study of the data-driven day-ahead hourly provincial load forecasting methods: From classical data mining to deep learning," *Renew. Sustain. Energy Rev.*, *119*, 109632.

[66] Bendaoud, N. M. M., Farah, N. (2020). "Using deep learning for short-term load forecasting," *Neural Comput. Appl.*, *32* (18), 15029–15041.

[67] Sadaei, H. J., e Silva, P. C. de L., Guimarães, F. G., Lee, M. H. (2019). "Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series," *Energy*, *175*, 365–377.

[68] Dong, X., Qian, L., Huang, L. (2017). "Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach," in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 119–125.

[69] Aurangzeb, K., Alhussein, M., Javaid, K., Haider, S. I. (2021). "A Pyramid-CNN Based Deep Learning Model for Power Load Forecasting of Similar-Profile Energy Customers Based on Clustering," *IEEE Access*, *9*, 14992–15003.

[70] Wu, K., Wu, J., Feng, L., Yang, B., Liang, R., Yang, S., Zhao, R. (2021). "An attention-based CNN-LSTM-BiLSTM model for short-term electric load forecasting in integrated energy system," *Int. Trans. Electr. Energy Syst.*, *31* (1), e12637.

[71] Cai, M., Pipattanasomporn, M., Rahman, S. (2019). "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," *Appl. Energy*, *236*, 1078–1088.

[72] Han, L., Peng, Y., Li, Y., Yong, B., Zhou, Q., Shu, L. (2018). "Enhanced deep networks for short-term and medium-term load forecasting," *IEEE Access*, *7*, 4045–4055.

[73] He, W. (2017). "Load forecasting via deep neural networks," *Procedia Comput. Sci.*, *122*, 308–314.

[74] Kim, J., Moon, J., Hwang, E., Kang, P. (2019). "Recurrent inception convolution neural network for multi short-term load forecasting," *Energy Build.*, *194*, 328–341.

[75] Choi, H., Ryu, S., Kim, H. (2018). "Short-Term Load Forecasting based on ResNet and LSTM," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 1–6.

[76] Chitalia, G., Pipattanasomporn, M. , Garg, V., Rahman, S. (2020). "Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks," *Appl. Energy*, *278*, 115410.

[77] Alhussein, M., Aurangzeb, K., Haider, S. I. (2020). "Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting," *IEEE Access*, *8*, 180544–180557.

[78] Kong, Z., Zhang, C., Lv, H., Xiong, F., Fu, Z. (2020). "Multimodal Feature Extraction and Fusion Deep Neural Networks for Short-Term Load Forecasting," *IEEE Access*, *8*, 185373–185383.

[79] Kalchbrenner, N., Oord, A., Simonyan, K., Danihelka, I., Vinyals, O.,

Graves, A., Kavukcuoglu, K. (2017). "Video pixel networks," in *Proceedings of the 34th International Conference on Machine Learning- 70*, 1771–1779.

[80] Fang, X., Yuan, Z. (2019) "Performance enhancing techniques for deep learning models in time series forecasting," *Eng. Appl. Artif. Intell.*, *85*, 533–542.

[81] Lahouar, A., Slama, J. B. H. (2015). "Day-ahead load forecast using random forest and expert input selection," *Energy Convers. Manag.*, *103*, 1040–1051.

[82] Breiman, L. (2001) "Random forests," *Mach. Learn.*, *45* (1), 5–32.

[83] Lyashevska, O., Harma, C., Minto, C., Clarke, M., Brophy, D. (2020). "Long-term trends in herring growth primarily linked to temperature by gradient boosting regression trees," *Ecol. Inform.*, *60*, 101154.

[84] James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An introduction to statistical learning, 112*. Springer.

[85] Chen, T., Guestrin, C. (2016). "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

[86] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y. (2017). "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 3146–3154.

[87] Vapnik, V. (1998). *Statistical learning theory*, 1st ed. Wiley-Interscience.

[88] Kuhn, M., Johnson, K. (2013). *Applied predictive modeling*, *26*, Springer.

[89] Hu, Z., Bao, Y., Xiong, T. (2014). "Comprehensive learning particle swarm optimization based memetic algorithm for model selection in short-term load forecasting using support vector regression," *Appl. Soft Comput.*, *25*, 15–25.

[90] Huang, G.-B., Zhu, Q.-Y., Siew, C.-K. (2006). "Extreme learning machine: theory and applications," *Neurocomputing*, *70* (1–3), 489–501.

[91] Pascanu, R., Mikolov, T., Bengio, Y. (2013). "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, 1310–1318.

[92] Hochreiter, S., and Schmidhuber, J. (1997). "Long short-term memory," *Neural Comput.*, *9* (8), 1735–1780.

[93] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv Prepr. arXiv1406.1078*, 2014.

[94] Binkowski, M., Marti, G., Donnat, P. (2018). "Autoregressive convolutional neural networks for asynchronous time series," in *International Conference on Machine Learning*, 580–589.

[95] Borovykh, A., Bohte, S., Oosterlee, C. W. (2017). "Conditional time series forecasting with convolutional neural networks," *arXiv Prepr.*

arXiv1703.04691.

[96] Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukvuoglu, K. (2016) "WaveNet: A generative model for raw audio.," *SSW*, 125.

[97] He, K., Zhang, X., Ren, S., Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

[98] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov D., Erhan, D.,Vanhoucke, V., Rabinovich, A. (2015) "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

CURRICULUM VITAE

Name Surname : İbrahim YAZICI

EDUCATION:

- **B.Sc.** : 2011, Kocaeli University, Industrial Engineering Department
- **M.Sc.** : 2015, ITU, Industrial Engineering Department

PROFESSIONAL EXPERIENCE AND REWARDS:

- Research Assistant in ITU- 2012/2021
- ITU VODAFONE FUTURE LAB R&D Operations Manager- 2021- Cont.
- TUBITAK Doctorate Scholar during PhD. studies

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Yazici, I,** Beyca, O.F., Delen, D. (2022), Deep-learning-based short-term electricity load forecasting: A real case application, Engineering Applications of Artificial Intelligence, *109*, 104645.