ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

Ph.D. THESIS

Uğur EJDER

DATA MINING BASED ON REGULARIZED CONVOLUTIONAL NEURAL NETWORK FOR TIME SERIES: FINANCIAL PREDICTION ALGORITHM.

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2022

ABSTRACT

Ph.D. THESIS

DATA MINING-BASED ON REGULARIZED CONVOLUTIONAL NEURAL NETWORK FOR TIME SERIES: FINANCIAL PREDICTION ALGORITHM

Uğur EJDER

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor	: Prof. Dr. Selma Ayşe ÖZEL
	Year: 2022, Pages: 105
Jury	: Prof. Dr. Selma Ayşe ÖZEL
	: Prof. Dr. Mutlu AVCI
	: Prof. Dr. Umut ORHAN
	: Assoc. Prof. Dr. Fatih KILIÇ
	: Assoc. Prof. Dr. Cihan ÇETİNKAYA

This thesis aims to design a generalizable distance-based moving average (DBEMA) method for predicting time series. In our study, we focused on a specific area of financial time series. In order to increase the performance of prediction accuracy, DBEMA was combined with features selected by Recursive Feature Elimination (RFE) by using Classification and Regression Tree (CART) estimators and sequential feature selection (SFS) by using Gradient Boosting Machine (GBM). Although many artificial neural networks (ANNs) have been applied to a number of time series predictions and modelling, convolutional neural networks (CNN) have not been used much for time series prediction directly in literature and are still open to improvement. For predicting the trend of time series with DBEMA, time series are defined in the form of different time-lagged moving average patterns to identify the relations between each of them. The distances between moving averages (MA) and changes in their positions towards each other are examined for predicting future trends of time series. First of all, time series are defined so as to cover different time lags of 9 days, 50 days and 200 days in exponential moving average (EMA) forms and the distances between each of them and positions between each of them are marked. To improve the performance of the distance-based moving average method, CART and GBM algorithms are used for selecting better financial features in with RFE and SFS models, respectively. The combination of distance-based features and selected financial features are converted into 2-D images which are then classified by CNN. According to the experimental results, the proposed algorithm, CNN-DBEMA, outperforms other classification techniques in literature.

Key Words: Distance-Based Features, Moving Average, Financial Time series Prediction, Convolutional Neural Network

ÖZET

DOKTORA TEZİ

ZAMAN SERİLERİ İÇİN VERİ MADENCİLİĞİ TABANLI DÜZENLİLEŞTİRİLMİŞ EVRİŞİMLİ SINIR AĞI: FİNANSAL TAHMİN ALGORİTMASI

Uğur EJDER

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Prof. Dr. Selma Ayşe ÖZEL
Yıl: 2022, sayfa: 105
Jüri : Prof. Dr. Selma Ayşe ÖZEL
: Prof. Dr. Mutlu AVCI
: Prof. Dr. Umut ORHAN
: Doç. Dr. Fatih KILIÇ
: Doç. Dr. Cihan ÇETİNKAYA

Bu tezin amacı, zaman serilerini tahmin etmek için genelleştirilebilir bir mesafe tabanlı hareketli ortalama (DBEMA) yöntemi tasarlamaktır. Çalışmamızda, finansal zaman serileri olarak adlandırılan belirli bir zaman serisi alanına odaklandık. Tahmin doğruluğu performansını artırmak için, mesafe tabanlı hareketli ortalama yöntemi, Sınıflandırma ve Regresyon Ağacı (CART) tahmin edicilerini kullanarak Özyinelemeli Özellik Eliminasyonu (RFE) ve Gradyan Artırma Makinesi (GBM) kullanarak sıralı özellik seçimi (SFS) ile elde edilen nitelikler ile birleştirildi. Zaman serisi tahmini ve modellemesi için birçok yapay sinir ağı (YSA) uygulanmış olmasına rağmen, literatürde evrişimli sinir ağları (CNN) zaman serisi tahmini için doğrudan çok fazla kullanılmamıştır ve halen geliştirilmeye açıktır. Zaman serilerinin trendini DBEMA ile tahmin etmek için, zaman serileri farklı gecikmeli hareketli ortalama desenleri şeklinde tanımlanarak aralarındaki ilişkiler meydana çıkarılır. Zaman serilerinin gelecekteki eğilimlerini tahmin etmek için hareketli ortalamaların (MA) aralarındaki mesafe ve birbirlerine göre konumlarındaki değişimler incelenir. Öncelikle zaman serileri 9 günlük, 50 günlük, 200 günlük olmak üzere değişik bekleme süreleri içerecek şekilde tanımlanır ve her birinin birbirine göre konumları ve mesafeleri işaretlenir. DBEMA yönteminin performansını artırmak için, CART ve GBM algoritmaları sırası ile RFE ve SFS nitelik seçim modellerinde daha iyi finansal özellik seçmek için kullanılmıştır. Mesafeye dayalı özellikler ve seçilen finansal özelliklerin birleşimi 2-boyutlu görüntülere dönüştürülür ve ardından CNN kullanılarak sınıflandırılır. Deneysel sonuçlara göre önerilen CNN-DBEMA algoritması literatürdeki diğer sınıflandırma tekniklerinden daha iyi performans göstermektedir.

Anahtar Kelimeler: Mesafe tabanlı nitelikler, Hareketli Ortalama, Finansal Zaman Serisi Tahmini, Evrişimli Sinir Ağı

EXTENDED SUMMARY

Financial time series prediction is a complex task due to the massive amount of data to be processed, frequent and nonlinear movements, and the variety of influencing factors, such as global economic conditions, government policies, exchange rates, inflation rates, company incomes, investor's moods. Furthermore, the efficient market hypothesis states that financial time series follow the random walk model, thus making it highly impossible to determine the direction of financial time series movements.

The academic studies depend widely on the macroeconomic parameters to predict the financial instruments, with rather little attention focused on the technical indicators that clients largely apply. Our study fills this gap by generating a novel distance-based moving average features and efficiently selecting technical indicators to improve the predictive ability. Technical indicators and oscillators are fundamental tools for researchers to discover financial time-series trends and develop strategies for making better investment decisions. This term is called technical analysis and researchers assume that financial time series move in trends and historical trends repeat themselves.

Technical analysis or tracing indicators have paid attention to historical movements of financial time series values and discovered hidden points so that researchers can predict trends of financial time series. Technical indicators improve decision skills to predict future trends of financial time series and they need to be searched for the best optimal rule to determine the highest return. Therefore, in our study, financial features generated by historical data are selected by two wrapper feature selection methods. These two wrapper methods can reach better feature subsets with the performance of the predefined classification methods. Recursive feature elimination (RFE) using classification and regression tree (CART) estimators is one of the approaches to determine a better feature subset. The other approach in determining a better feature subset is the sequential feature selection (SFS) using

Gradient Boosting Machine (GBM). Selected technical indicator features by the RFE and SFS methods are merged and used in this study.

The Moving Average (MA) crossover method is one of the famous technical analysis devices researchers use in time series analysis. This technique is based on determining short and long time moving average periods and identifying crossover to each other. The decision of which moving averages must be selected is an important topic for researchers. Nonetheless, most researchers base their selection of these parameters on recommendations that are not appropriate for a specific market. Accordingly, the MA crossover method is suitable for offering significant returns for the time series data. MA crosses to each other strategy identify the trend of time series. However, this does not mean that there is a generalization that this strategy always works this way. In some cases, some situations break this habit. In order to overcome this problem, our forecasting of success has increased significantly with the new model, which is defined distance-based exponential moving average (DBEMA) methodology. The reason for choosing exponential moving averages (EMA) is because they respond more quickly to changing situations. However, it is possible to work with the desired moving averages according to the situations with the method we have offered. In this study, daily value was used in order to calculate moving averages for a period. Moving averages used have 9, 50, and 200 days-period values. The shortest period EMA is defined as EMA9 (exponential moving average for 9 days period) and the longest period is defined as EMA200 (exponential moving average for 200 days period). In literature, the system usually works as follows: When the EMA9 short-term moving average crosses above the EMA200 long-term moving average, this means a buy signal is produced by the system but sometimes this situation can be a trap. A novel distance-based moving average method system can detect the proper intention of the trend. In another case, the EMA9 short-term moving averages crosses below the EMA200 long-term moving averages, which means the system produces a sell signal, but this can be a trap case. The distance-based approach has developed a new system to overcome the trap problem. In this method, the system tries to classify four types of cases which are Earning class (EC), Losing Class (LC), Earning Trap Class (ETC), and Losing Trap Class (LTC) (Ihab, 2009). The new system focuses on technical indicators and the distance between the moving averages in this situation.

The distance-based system is entirely parametrically designed. When the EMA9 short-term moving average crosses above the EMA200 long-term moving average, the system produces at least *x%* earnings within *t* days, where *x* is *earnings_ratio* (e.g., *x*=15), and *t* is *with_in_day* variable. This means that this case is “Earning Class”. When the EMA9 short-term moving average crosses above the EMA200 long-term moving average and the system does not produce at least *x%* earnings within *t* days, this case is “Earning Trap Class”. When the EMA9 short-term moving average crosses below the EMA200 long-term moving average, and the system produces at least *y%* loss within *t* days, where *y* is *losing _ratio* and *t* is *with_in_day* variable, this means that this case is “Losing Class”. When the EMA9 short-term moving average crosses below the EMA200 long-term moving average, and the system does not produce at least *y%* loss within *t* days means, this case is “Losing Trap Class”. Producing *x%* earning or *y%* losing within *t* days period are parametric variables and they are determined by the user of the system. As a result, the proposed system is adjustable, and it can be determined for any user-defined earning/losing ratio for a user-defined period. The class labels are defined according to the above description.

After defining the system's features and distance-based moving averages, the machine learning algorithms are applied. In literature, deep learning methods have started appearing in financial studies. In literature, the application of convolutional neural network (CNN) to financial prediction models has been limited. CNN is a part of artificial neural networks that have become a key role in various computer vision tasks. Most of the CNN based systems in the literature have focused on computer vision and image processing. However, its use in the financial time series analysis domain has been limited. In this study, a CNN based system is designed to utilize the efficiency of a deep convolutional neural network for a robust decision support system. For this purpose, it converts financial time series into 2-D image-like data. Each image has 18 different financial technical indicators selected from SFS and RFE algorithms and additional 6 different distance-based moving average features with 75 days rows to illustrate each image-like data. The reason for choosing 75 days period was determined by our experimental analysis. Our system parameters are listed as follows: Our original image-like input size is 75 x 24, two convolutional layers with sizes 32 x 75 x 24 and 32 x 35 x 10, two max-pooling layers having sizes 32 x 71 x 20 and 64 x 33 x 8, three dropout values (0.3, 0.3, 0.3) are used for avoiding overfitting. The convolutional layers are connected to two fully connected layers (256,128). In the first convolution layer, we use 32 filters on a 75 x 24 input 2D array with the size of 5x5, and a stride is 1, and we use 64 filters on the second layer. The activation function applied at this layer is ReLU. At the end of the method, the output layer passes through the softmax function with four neurons.

The performance of our proposed CNN-DBEMA model has been compared to different kinds of methods. The categories of the methods used for comparison are financial methods, machine learning methods, and one of the recently proposed CNN-TA method (Sezer. et al, 2018). Therefore our proposed model CNN-DBEMA is compared to the other methods, which include the Linear Regression model (LR), Gaussian Naive Bayes Model (NB), K-Nearest Neighbor Classifier (KNN), Random Forests (RF), Multi-Layer Perceptron (MLP), Bagged Trees Regressor (BAG), Relative Strength Indicator Strategy (RSI), and new novel model CNN-TA. Three types of data sets were used for the financial time series trend forecasting tests between the benchmarking models. At first, the system tested the forecasting accuracy with the first type of data set, which includes crossover (Last) points of data. In accuracy of forecasting results, our proposed CNN-DBEMA model reaches second best score among the benchmarking models. Even though CNN-DBEMA gets the second-highest accuracy score, the novel CNN-DBEMA model reaches a better specificity and sensitivity score. In order to ensure that the proposed CNN-

DBEMA model is evaluated accurately, we have tested the second type of data set that includes the average value of the last 75 days of the financial time series. The performance of the proposed model versus benchmarking models yielded the best accuracy result, which contains the average value of the last 75 days of stocks. The last approach towards verifying the accuracy performance of our proposed CNN-DBEMA model versus benchmarking models is to include all data points of input matrices. When the system uses second and third types of data CNN-DBEMA model predicted better results compared to the other models.

In this study, we aim to design a new novel CNN-DBEMA method for stock trend forecasting by considering both the new concept of distance-based exponential moving averages (DBEMA) and technical indicators. The most important innovation is discovering that the distances between the exponential moving averages give us a prediction for the financial time series. At the same time, these new distance-based features will be used as new parameters in other areas such as pattern recognition, signal processing, weather forecasting, and earthquake prediction and will raise the success of the estimation.

The CNN-DBEMA model will be developed with possible improvements for the future progress of the trend prediction method. Firstly, various technical indicators will be tried to generate a better prediction model. Secondly, different images can be produced by different periods of moving averages. We will also apply our model to different types of dataset that contains different financial assets

GENİŞLETİLMİŞ ÖZET

Finansal zaman serisi tahmini, işlenecek çok büyük miktarda veri, sık ve doğrusal olmayan hareketler ve küresel ekonomik koşullar gibi etkileyen faktörlerin çeşitliliği, hükümet politikaları, döviz kurları, enflasyon oranları, şirket gelirleri, yatırımcıların ruh halleri nedeniyle karmaşık bir iştir. Ayrıca, etkin piyasa hipotezi, finansal zaman serilerinin rastgele yürüyüş modelini takip ettiğini ve dolayısıyla finansal zaman serisi hareketlerinin yönünü belirlemeyi oldukça imkansız hale getirdiğini belirtir.

Akademik çalışmalar, finansal enstrümanları tahmin etmek için büyük ölçüde makroekonomik parametrelere bağlıdır ve müşterilerin büyük ölçüde uyguladığı teknik göstergelere oldukça az dikkat gösterilir. Çalışmamız, mesafeye dayalı yeni bir hareketli ortalama oluşturarak ve tahmin yeteneğini geliştirmek için etkili bir yöntemle teknik göstergeleri seçerek bu boşluğu dolduruyor. Teknik göstergeler ve osilatörler, araştırmacıların finansal zaman serisi trendlerini keşfetmeleri ve daha iyi yatırım kararları almak için stratejiler geliştirmeleri için temel araçlardır. Bu terime teknik analiz denir ve araştırmacılar finansal zaman serilerinin trendler halinde hareket ettiğini ve tarihsel trendlerin kendilerini tekrar ettiğini varsaymaktadır.

Teknik analiz veya izleme göstergeleri, finansal zaman serisi değerlerinin tarihsel hareketleri vasıtasıyla gizli noktaları keşfederek araştırmacıların finansal zaman serilerinin eğilimlerini tahmin edebilmelerini sağlar. Teknik göstergeler, finansal zaman serilerinin gelecekteki eğilimlerini tahmin etmek için karar verme becerilerini geliştirir ve en yüksek getiriyi belirlemek için en iyi optimal kuralın aranması gerekir. Bu nedenle, çalışmamızda tarihsel verilerle oluşturulan finansal özellikler, iki sarmalayıcı nitelik seçimi yöntemiyle seçilir. Olası özellik alt kümelerinin uygunluğunu belirlemek için iki sarmalayıcı nitelik seçimi algoritması kullanılır. Bu iki sarmalayıcı yöntem, önceden tanımlanmış sınıflama yöntemlerinin performansı ile daha iyi özellik alt kümelerine ulaşabilir. Sınıflandırma ve regresyon ağacı (CART) tahmin edicilerini kullanan özyinelemeli özellik eliminasyonu (RFE), daha iyi bir özellik alt kümesi belirlemeye yönelik bir sarmalayıcı yaklaşımdır. Daha iyi bir özellik alt kümesi belirlemeye yönelik diğer yaklaşım da, Gradyan Artırma Makinesi (GBM) kullanan sıralı özellik seçimidir (SFS). Temel olarak, SFS, ilk özellik boyut alanını, alt küme özellik alanına azaltmak için kullanılır. Öznitelikleri seçme, RFE ve SFS sonuçları birleştirilerek elde edilmiştir.

Hareketli Ortalama (HO) üstüne geçme yöntemi, araştırmacıların zaman serisi analizinde kullandıkları ünlü teknik analiz yöntemlerinden biridir. Bu teknik, kısa ve uzun süreli hareketli ortalama periyotlarının belirlenmesine ve birbirinin üzerine geçişin belirlenmesine dayanmaktadır. Hangi hareketli ortalamaların seçilmesi gerektiğine dair karar araştırmacılar için önemli bir konudur. Bununla birlikte, çoğu araştırmacı, bu parametrelerin seçimini belirli bir pazar için uygun olmayan önerilere dayandırır. Buna göre, HO üstüne geçme yöntemi, zaman serisi verileri için önemli getiriler sunmak için uygundur. HO birbirinin üstüne geçme stratejisi zaman serilerinin trendini belirler. Ancak, bu stratejinin her zaman bu şekilde çalıştığına dair bir genelleme olduğu anlamına gelmez. Bazı durumlarda, bazı olaylar bu genellemeyi bozar. Bu sorunu aşmak için mesafe tabanlı hareketli ortalama metodolojisi olarak tanımlanan yeni model ile başarı öngörümüz önemli ölçüde artmıştır. Bizim çalışmamızda, üstel hareketli ortalamalar (EMA) ile yeni bir mesafe tabanlı yöntem tasarlandı. Üstel hareketli ortalamaların seçilmesinin nedeni, değişen durumlara daha hızlı tepki vermeleridir. Ancak sunduğumuz yöntemle duruma göre istenilen hareketli ortalamalar ile çalışmak mümkündür. Bu çalışmada hareketli ortalamaların hesaplanmasında dönem olarak günlük değer kullanılmıştır. Hareketli ortalamalar 9, 50 ve 200 gün dönemlik değerler için kullanıldı. En kısa dönem EMA, EMA9 (9 günlük hareketli ortalama) olarak ve en uzun dönem EMA200 (200 günlük hareketli ortalama) olarak tanımlanmıştır. Literatürde genellikle sistem şu şekilde çalışır: EMA9 kısa vadeli hareketli ortalama, EMA200 uzun vadeli hareketli ortalamanın üzerine çıktığında, bu sistem tarafından bir satın alma sinyali üretildiği anlamına gelir ancak bazen bu durum bir tuzak da olabilir. Uzaklığa dayalı yeni hareketli ortalama yöntem sistemi, trendin doğru amacını tespit edebilir. Başka bir durumda, EMA9 kısa vadeli hareketli ortalaması, EMA200 uzun vadeli hareketli ortalamasının altına düşerse, bu da sistemin bir satış sinyali ürettiği anlamına gelir, ancak bu bir tuzak da durumu olabilir. Mesafeye dayalı bir yaklaşım, tuzak sorununun üstesinden gelmek için yeni bir sistem geliştirmiştir. Bu yöntemde sistem dört tip vakayı sınıflandırmaya çalışır. Bunlar Kazanç sınıfı (EC), Kayıp Sınıfı (LC), Kazanç Tuzağı Sınıfı (ETC) ve Kayıp Tuzağı Sınıfı (LTC) olarak isimlendirilmiştir. Önerilen sistem, bu durumda teknik göstergelere ve hareketli ortalamalar arasındaki mesafeye odaklanır.

Mesafe tabanlı sistem tamamen parametrik olarak tasarlanmıştır. EMA9 kısa vadeli hareketli ortalama, EMA200 uzun vadeli hareketli ortalamanın üzerine çıktığında, sistem t gün içinde en az % x kazanç üretirse, burada x kazanç_oranıdır (örneğin, x=15), bu durum Kazanan Sınıf anlamına gelir. EMA9 kısa vadeli hareketli ortalama, EMA200 uzun vadeli hareketli ortalamanın üzerine çıktığında, ve sistem t gün içinde en az % x kazanç sağlayamazsa bu durum Tuzak Kazanç Sınıfıdır. EMA9 kısa vadeli hareketli ortalama, EMA200 uzun vadeli hareketli ortalamanın altına düştüğünde, ve sistem t gün içinde en az %y kayıp üretmiyorsa, bu durum Tuzak Kayıp Sınıfı demektir. t gün içerisinde %x kazanç veya %y kayıp parametrik değişkenlerdir ve sistem kullanıcısı tarafından belirlenir. Sonuç olarak, önerilen sistem ayarlanabilir ve kullanıcı tanımlı herhangi bir dönem için kullanıcı tanımlı herhangi bir kazanç/kayıp oranı için belirlenebilir. Etiketleme sürecinin tanımlanması yukarıdaki tanımlamaya göre tasarlanmıştır.

Sistemin özelliklerini ve mesafe bazlı hareketli ortalamaları tanımladıktan sonra, makine öğrenmesi algoritma sürecini tasarlama çalışmaları geliştirilmiştir. Literatürde finansal çalışmalarda derin öğrenme yöntemleri yer almaya başlamıştır. CNN, çeşitli bilgisayarlı görü görevlerinde kilit rol oynayan yapay sinir ağlarının bir parçasıdır. Literatürdeki CNN sisteminin çoğu bilgisayarla görme ve görüntü işlemeye odaklanmıştır. Ancak finansal zaman serisi analizi alanında kullanımı sınırlı kalmıştır. Bu çalışmada, sağlam bir karar destek sistemi için derin evrişimli sinir ağının verimliliğinden yararlanmak amaçlanmıştır. Bu amaçla finansal zaman serileri 2 boyutlu görüntü benzeri verilere dönüştürülür. Her bir görüntü benzeri veriyi göstermek için SFS ve RFE algoritmalarından gelen 18 farklı finansal teknik gösterge ile 6 farklı mesafeye dayalı hareketli ortalama teknik göstergesi kullanılır. Bu özellikler geriye doğru 75 günlük veriyi içermektedir. 75 gün seçilmesinin nedeni deneysel olarak belirlenmiştir. Sistem parametreleri şu şekilde seçilmiştir: Orijinal görüntü benzeri girdi boyutu 75 x 24, iki evrişim katmanı boyutları 32 x 75 x 24 ve 32 x 35 x 10, iki maksimum havuzlama katmanı sırasıyla 32 x 71 x 20, 64 x 35 x 10 boyutları, aşırı öğrenmeyi önlemek için üç bırakma (0,3,0,3,0,3) değerleri kullanılır. Evrişimsel katmanlar, tam olarak bağlı iki katmana (256,128) bağlanır. Birinci evrişim katmanında, 5 x 5 boyutunda ve adımı 1 olan çerçeve ile 75 x 24 girişli 2D dizide 32 filtre ve ikinci katmanda 64 filtre kullanıyoruz. Bu katmanda uygulanan aktivasyon fonksiyonu RELU 'dur. Yöntemin sonunda çıktı katmanı 4 nöron ile softmax fonksiyonundan geçer.

Önerilen CNN-DBEMA modelimizin performansı farklı türdeki yöntemlerle karşılaştırılmıştır. Bu kategoriler finansal yöntemler, makine öğrenme yöntemleri ve yakın zamanda önerilmiş CNN-TA yöntemidir. Önerilen modelimiz CNN-DBEMA'nın değerlendirilmesi, Lineer Regresyon modeli (LR), Gauss Naive Bayes Modeli (NB), K-En Yakın Komşu Sınıflandırıcısı (KNN), Rastgele Ormanlar (RF), Çok Katmanlı Algılayıcı (MLP), Torbalı Ağaçlar Regresörü (BAG), Göreceli Güç Göstergesi Stratejisi (RSI) ve CNN-TA modelleri ile karşılaştırılmıştır. Kıyaslama modelleri arasındaki finansal zaman serisi trend tahmin testleri için üç tip veri seti kullanılmıştır. İlk başta, sistem üste geçme (Son) veri noktalarını içeren ilk veri seti türü ile tahmin doğruluğunu test etmiştir. Tahmin sonucunun doğruluğunda, önerilen CNN-DBEMA modelimiz, kıyaslama modelleri arasında en yüksek ikinci puana ulaşmıştır. CNN-DBEMA en yüksek ikinci doğruluk puanını ikinci sırada alsa da, CNN-DBEMA modeli daha iyi bir kesinlik ve duyarlılık puanına ulaşmıştır. Önerilen CNN-DBEMA modelinin doğruluğunun değerlendirilmesini sağlamak için, son 75 günlük finansal zaman serisinin ortalama değerini içeren ikinci tür veri setini test ettik. Önerilen modelin kıyaslama modellerine karşı performansı, son 75 günlük hisse senetlerinin ortalama değerini içeren veri setinde en iyi doğruluk sonucunu elde etmiştir. Önerdiğimiz CNN-DBEMA modelimizin, kıyaslama modellerine kıyasla doğruluk performansını doğrulamaya yönelik son yaklaşım giriş matrislerinin tüm veri noktalarını içermektedir. İkinci ve üçüncü tip veriler kullandığında CNN-DEMA modeli, test edilen modeller arasında en iyi tahmini yapmıştır.

Bu çalışmada, hisse senedi trend tahmini için hem uzaklığa dayalı üstel hareketli ortalamalar (DBEMA) hem de teknik göstergeleri göz önünde bulundurarak yeni bir CNN-DBEMA yöntemi tasarlamayı amaçladık. En önemli yenilik, üstel hareketli ortalamalar arasındaki mesafelerin finansal zaman serilerinin tahmini için kullanılabileceğinin gösterilmesi oldu. Sonraki çalışmalarda, bu yeni önerilmiş mesafe tabanlı özelliklerin örüntü tanıma, sinyal işleme, hava tahmini ve deprem tahmini gibi diğer alanlarda, yeni parametreler olarak kullanılması ve tahmin başarısına etkisi araştırılabilir.

CNN-DBEMA modeli, trend tahmin yönteminin gelecekteki çalışmaları için olası iyileştirmelerle geliştirilecektir. İlk olarak, daha iyi bir tahmin modeli oluşturmak için çeşitli teknik göstergeler denenecektir. İkinci olarak, farklı hareketli ortalama periyotları ile farklı görüntüler üretilebilir. Modelimizi, farklı finansal varlıkları içeren farklı veri kümesi türlerinde de uygulanabilir.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Prof. Dr. Selma Ayşe ÖZEL for her valuable support, patience, guidance, and motivation in completing this research work.

I wish to extend my special thanks to Prof. Dr. Umut ORHAN, and Prof. Dr. Mutlu AVCI for their valuable help, support, and suggestions.

I am also thankful to Assoc. Prof. Dr. Cihan ÇETİNKAYA, Assoc. Prof. Dr. Utku GÜGERCİN, Assoc. Prof. Dr. Fatih KILIÇ, and Buse Melis ÖZYILDIRIM for their support and motivation.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

LIST OF ABBREVIATIONS

ANN : Artificial Neural Networks

ARIMA : Autoregressive Integrated Moving Average

BAG : Bagged Trees Regressor

BIST : Borsa Istanbul

CART : Classification and Regression Tree

CNN : Convolutional Neural Networks

DBEMA : Distance-based Exponential Moving Average

DL : Deep Learning

EC : Earning Class

EMA : Exponential Moving Average

ETC : Earning Trap Class

GBM : Gradient Boosting Machine

KNN : K-Nearest Neighbors

LC : Losing Class

LR : Linear Regression

LSTM : Long Short Term Memory

LTC : Losing Trap Class

MFI : Money Flow Index

ML : Machine Learning

MLP : Multi-Layer Perceptron

RFE : Recursive Feature Elimination

RSI : Relative Strength Index

SFS : Sequential Feature Selection

SMA : Simple Moving Average

1. INTRODUCTION

Time series analysis is a particular way of analyzing a sequence of various data points gathered over a given period of time. The purpose of analyzing time series is to understand historical dataset and solve several time-based problems. The prediction of the future manner of the financial time series is a vital action in the realization of risk management, trend of financial instruments, interest rates and portfolio assignment. Researchers have examined the development of prediction models to forecast the future of unusual trends. Linear and nonlinear financial time series have been presented to face up to a general financial time series. Autoregressive integrated moving average (Box, Jenkins, and Reinsel, 1994), and statistical approaches like GARCH (Durham, 2007) are among the traditional methods that have been used for financial time series analysis.

Recently, financial time series prediction became definitely one of the most popular topics for applying artificial intelligence techniques for investors in academia and the financial field because of its wide implementation areas and significant effect. Financial time series prediction is one of the most challenging issues among forecasting problems because of its noisy and volatile nature. Machine Learning (ML) scientists have produced several methods, and a lot of studies have been published accordingly. As such, a substantial amount of research exists including ML studies on financial time series prediction. As an example, stock price index has been predicted with SVM in (Kim, 2003). Artificial neural networks (ANNs) are used as another encouraging approach, concerning statistical models, for financial time series forecasting (Ma et al., 2015). Lately, Deep Learning (DL) methods have been introduced in the field with results that remarkably exceed their traditional ML models. However, there is a big interest in expanding methods for financial time series prediction, and there is a skill gap in improved studies that entirely focus on DL for financial time series. In literature, DL architectures use directly related feature representations from the raw data and are modelled (Chong et al., 2017).

Therefore, the aim of this study is to offer an all-inclusive methodology in an entire-time series prediction domain such as weather data, rainfall measurements, temperature readings, quarterly sales, financial instrument prices, industry forecasts, interest rates. Hence, we do not only produce solutions for financial implementation areas, but we also apply our times series prediction methodology based on DL models, especially Convolutional Neural Networks (CNNs). We tried to create a vision for future studies and call attention to opportunities for the utility of researchers. Prediction of financial time series arrangement is to set the purposes appropriately and design in accordance with their own style and predict the trend of financial time series to accomplish the goal. One significant technique for successful prediction is to set prediction goals over time. For this purpose, there are different term periods that are short-term, medium-term, and long-term. The length of the terms is based on the selected time-period input variables that are defined by the systems. The time period is defined as 1-hour, 4-hours, 1-day, and 1-week. It is adjustable variable for the designed system which is selected for desired purposes.

The distance-based exponential moving average (DBEMA) methodology is designed with 9-days, 50-days, and 200-days period input variables in our study. Each period is defined as 1-daily value. Convolutional Neural Network - Distance-Based Exponential Moving Average (CNN-DBEMA) focuses on features that are created with 1-day period input variables. The CNN-DBEMA input variables consist of two groups. The first group is created by financial indicators, and the second group is created by a novel distance-based moving average variable. The first group of features are selected by the union of two wrapper-based feature selection algorithms. The first wrapper algorithm is sequential feature selection (SFS) using Gradient Boosting Machine Algorithm (GBM). In literature, experimental results show that GBM outperforms the other benchmark models (Levantesi and Zacchia, 2021). SFS uses GBM that learns which features are most enlightening at each time step, selecting the next feature depending on the already adopted features. The second wrapper algorithm Recursive Feature Elimination (RFE) is used with classification and regression tree (CART) classifier to reduce number of features and associated costs during classification. In addition to this, choosing CART outperforms the other benchmark models. SFS using GBM and RFE using CART take 11 features and 18 features, respectively. The combined result of two wrapper algorithms, as well as our proposed distance based exponential moving average features are used in the proposed model for financial time series prediction.

Deep learning has revealed a magnificent achievement in most complicated tasks such as image processing, natural language processing, cyber security, etc. (Yasir and Croock, 2020). Convolutional neural networks (CNNs) are part of deep learning algorithms, that are appropriate for automatic feature extraction and prediction (LeCun et al., 2015). In many fields, such as image processing and natural language processing, DL methods have been proved that they are able to create step by step effective complicated features from raw data or simpler parameters (He et al., 2016; LeCun et al., 2015). Extracting features that are informative enough to make predictions is a fundamental challenge as the behavior of equity markets is more complex, nonlinear, and noisy, and DL seems like a promising approach in this regard (Hoseinzade and Haratizadeh, 2019). Therefore, in this improvement of the thesis, we propose to apply CNN to the combination of DBEMA-based features and technical indicator based features that are selected by SFS using CART and RFE using CART to make financial time series analysis. One of the other important points is that the sequence of the features is not random. To perform better results, similar behaving features are clustered closely for uniform and effective time series image representation. For the analysis of the financial time series image, CNNs are applied.

1.1. The Aims and Objectives of this Thesis

Financial time series prediction has been growing into an alluring subject in so many aspects due to its wide performing areas and considerable effect. With the increase in financial instruments' ability to make decisions in the markets, it is aimed to develop autonomous and intelligent expert systems. In line with this aim, different types of machine learning techniques have been used for financial time series prediction. However, there are many studies on time series and the amount of implementation of CNNs for the financial time series prediction seems to be quite a few. The reason for having the insufficient number of application of CNNs for financial time series analysis is the challenge in the representation of time series as an image-like 2-D array and labelling complexity. Hence, we aim to improve time series prediction results and suggest a generalized solution for all types of time series. It is noticed that financial time series prediction has a comprehensive usage and substantial impact on the future plans and behavior of the financial areas (Kirisci et al., 2022; Thomas and Christopher, 2018).

From this point of view, in this thesis, it is proposed to utilize financial technical indicators effectively for better prediction (Thomas and Christopher, 2018), limited use of deep learning methods (Kraussa et al., 2017), offering new distance-based moving average methods, offering generalized solution architecture for all kind of times series. For this purpose, a novel distance-based exponential moving average (DBEMA) features are proposed to analyze financial time series with CNN model to make stable decisions and provide high profits. Deep neural network architecture (CNN) to model nonlinear relationships between dependent and independent variables is better than an artificial neural network. That is why we apply CNN in this study. Deep neural network architectures are fed with two groups of input variables. The first group is technical indicators calculated from raw data of financial instruments and the second group is a novel distance-based moving average indicators.

The performance of the proposed model was compared with state-of-art methods using a dataset obtained from Borsa Istanbul data.

1.2. Contributions of this Thesis

Studies included in this thesis aim to cover the generalized solution model approach to all kinds of time series and increase its prediction accuracy. To this end, a novel distance-based exponential moving average methodology is designed and applied to financial time series data to show how to increase the prediction performance. It will be mentioned how to use two-dimensional time series data more efficiently, which are rarely used in CNN networks in the literature.

In the literature, there are generally buy, sell, and hold signals for financial time series analysis (Gunduz et al., 2017; Sezer and Ozbayoglu, 2018). Especially, Earning Trap Class, and Losing Trap Class are not used. In our approach, we also considered trap classes and brought a new perspective. We analysed financial time series data as 4 class classification problem.

It is aimed to optimize the technical analysis input parameters used in the CNN, by applying feature selectors SFS using RF model and RFE using CART model.

The thesis primarily paid attention to a novel distance-based moving average method, and its advantages in financial time series estimation are investigated. We also explore the potential of CNNs in financial time series prediction and contribute to the literature by comparing CNN model with other methods used in financial time series analysis. Due to the complex and dynamic nature of the stock market, it is difficult to predict the value of the financial time series. The other purpose of this thesis is to support investors increase their profitability.

1.3. Outline of the Thesis

This thesis is organized as follows:

In Section 2, the related studies in the literature are reviewed and general information is given about them. Technical and fundamental models, machine learning models, deep learning and convolutional neural network models, and machine learning models in BIST studies in the literature are summarized.

Section 3 covers materials for this thesis that are distance-based moving average methodology, CNN-DBEMA model, and benchmark models algorithms. This section explains details about the dataset used; buy, sell, and trap infrastructure rules; selection of appropriate technical indicator features using two wrapper algorithms for the CNN architecture. Different types of input parameters are generated to ensure the accuracy of the comparison.

In Section 4, the experimental results of the algorithms are analysed and compared with the benchmark methods in the literature. The financial results of the proposed CNN-DBEMA model are examined.

The last section covers a comparison of the proposed methods with benchmark methods, the discussions on results, and future works.

2. LITERATURE REVIEW

This section introduces the main approaches used in technical and fundamental analysis, machine learning methods, and classification methods for financial time series analysis. Then, we resume the latest stock market forecasting studies. We include the latest market forecasting studies on Borsa Istanbul (BIST). Besides, we also summarize the latest deep learning techniques in forecasting financial time series.

2.1. Technical and Fundamental Approach for Financial Time Series

Fundamental and technical analysis are two major approaches to time series analysis. Researchers and investors use both techniques to investigate and predict future time series trends. The technical analysis pays attention to historical data on financial instruments, whereas fundamental analysis focuses on economic conditions to forecast the intrinsic values of financial instruments

The random walk idea follows the hypothesis that financial time series movements are explained randomly, and it is unstable (Malkiel 1973). The financial time series is described as very difficult to predict because of its uncertain and nonlinear environment (Abu-Mostafa and Atiya, 1996). It is impossible to forecast financial time series values based on random walk theory, but using the technical and fundamental techniques, and progress in machine learning and classification methods made financial forecasting possible.

Applying traditional financial time series methods can be achieved by discovering the mathematical and statistical parameters such as moving average, variance, standard deviation, minimum, maximum, covariance, correlation, and autocorrelation (Rodolfo et al., 2017). A purpose of investigating stock market inefficiencies purposes to explore the effectiveness of investment strategies using the three most common technical indicators that are Moving Average (MA), Moving Average Divergent Convergent (MACD), and Relative Strength Index (RSI), considering stock market situations, with and without trading costs and transaction fees (Huy and Cuong, 2018). The commodity price changes can be predicted with technical indicators. Technical indicators are better predictors than economic indicators, and their prediction performances are not altered by conditions. In addition, the investigation demonstrates that technical indicators also perform better than economic parameters in predicting the density of commodity price movements (Wang et al, 2020). Technical analysis has been used by traders, analysts, researchers, and investors that allows for predicting future movements in financial instruments by concentrating on past market data, generally price and volume (Park, and Irwin, 2007; Menkhoff and Taylor, 2007).

A technical analysis strategy of the Greek Stock Market study used most frequently applied technical indicators such as MACD, RSI, and stochastic approaches for using patterns and candlestick charts. The results show strong technical trading rules (Rousis and Papathanasiou, 2018). Another study aims to define which indicators are more efficient in presenting more accurate buy and sell signals on the LQ45 index by using the oscillator indicator MACD, Bollinger Band and RSI (Pramudya and Ichsani, 2020). Autoregressive integrated moving average (ARIMA), regression analysis, curve fitting, and autoregressive moving average (ARMA) are the technical analysis systems that are generally used to explore and forecast financial time series in literature (Box et al., 2015).

Some of the studies for modeling financial time series include the prediction of variables from a proposed time-series model, such as Linear Dynamical Systems (LDS) (Luenberger 1979), the popular Hidden Markov Model (HMM) (Rabiner, 1986), and autoregressive (AR) models (Lütkepohl, 2005). A novel distributed approach concurrently clusters and fits AR model groups of similar individuals (Benny and Ian, 2022). The AR model is one of the primary methods for classifying time series. It is a simple method, but it accepts that the stationary type time series has some disadvantages in categorizing them (Kini and Sekhar, 2013).

2.2. Machine Learning Approach for Financial Time Series

In the literature, many machine learning based approaches have been proposed for time series estimation. To predict future movements in financial time series Neural Networks (NN) (Kayal, 2010; Vui et al., 2013; Persio and Honchar, 2016) and Support Vector Machines (SVM) (Markovic et al., 2014; Sapankevych and Sankar 2009) have been used. To offer researchers with better prediction results, a revised modeling procedure is suggested which aims to increase the function of the k-nearest neighbors (KNN) method in financial time series forecasting. The main idea is to join the aggregate empirical mode decomposition (EEMD) method with Multidimensional k-nearest neighbors time series estimation with invariance (MKNN–TSPI). They present an example of two-dimensional stock series, the revised method is applied to predict the opening and closing prices of NAS, DJI, S&P 500, Russell 2000, and other stocks concurrently (Guancen et al., 2021).

It is noticeable that multiple classifiers can usually outperform a single classifier (Dietterich, 2000). It is seen in another study that regression models are used to predict the future movements of financial time series (Chen et.al, 2015; Jiang et al., 2020; Zhang et al., 2020). Moreover, other studies try to predict the future movements of financial time series with classification methods (Cagliero et al., 2020; Ananthi and Vijayakumar, 2020). However, to make accurate forecast on financial time series, researchers are more focused on the future movements of financial time series. Many researchers attempt to design artificial neural networks (ANN) to forecast stock market index values. As an example, in (Guresen et al., 2011), researchers define the performance of dynamic ANN and the neural network models especially multilayer perceptron (MLP) to predict the NASDAQ stock index. The other study (Sezer et al., 2017) utilizes technical indicators as input variables and the ANN classifier is set on the most significant input variables to make predictions. Also, it devises an ANN that uses inputs of the financial technical indicators (i.e., RSI, MACD, Williams%R) to predict Dow30 stock prices (Sezer et al., 2017). Moreover, hybrid systems have been proposed to investigate financial time series data. (Wang et al., 2016) proposed to combine principal component analysis (PCA) and brain storm optimization (BSO) for a hybrid SVR model to forecast stock prices.

In another model, 20 different technical indicators are used as input variables for the model. The system merged with SVR, PCA, and BSO to use technical indicators for forecasting. The application of SVM in financial time series forecasting is compared with the multilayer back-propagation (BP) neural network and the regularized radial basis function (RBF) neural network. The diversity in the performance of SVM with respect to the free inputs is analyzed experimentally (Tay, 2003). Another proposed aggregate model of machine learning system consists of rule-based and MLP evolutionary algorithms that offer sell and buy signals in stock prices (Mabu et al., 2015). The performance of ensemble models is evaluated by random forest, AdaBoost, kernel factory, and classifier models (SVM, ANN, KNN, and logistic regression) to predict financial time series movements in the stock market (Ballings et al., 2015).

In stock market studies, ensemble learning models have been used instead of a single classifier (Patel et al., 2015). Patel et al. (2015) predicted movement of Indian stock markets and compared the classification performances of ANN, SVM, Random Forest, and Naive Bayes algorithms. Between 2003 and 2012 years daily open, high, low, and close values are used for creating technical indicators. The Random Forrest model performed better than the other three models.

Rivero and Rodriguez (2010) examined the causality between S & P 500, FTSE 100, and Nikkei 225 index with Gradient Boosting Algorithm during the unstable price changes. They discovered the valuable information in improving the forecast of index income.

2.3. Deep Learning and Convolutional Neural Network Approach for Financial Time Series

Deep learning models are quite popular and DL models are at different types that are namely deep neural network (DNN), recurrent neural network (RNN), long short term memory (LSTM), restricted Boltzmann machines (RBMs), convolutional neural network (CNN), deep belief networks (DBN). Deep learning is a machine learning technique that is appropriate for automatic feature extraction and forecasting (LeCun et al., 2015). In literature, CNNs, DBN, and RBM are generally used for image processing and classification. LSTM and RNN architectures specifically are designed for analyzing sequential data, time-series data, speech recognition, and natural language processing. Moreover, CNNs are mainly used in image processing, classification, and recognition processes (Krizhevsky et al., 2012; Karpathy et al., 2014; Lawrence et al., 1997).

Chong et al. (2017), aim to offer an extensive evaluation of both the advantages and disadvantages of deep learning algorithms for stock market analysis and forecasting. High-frequency intra-day stock returns are used as input values and explore the influences of three unsupervised feature extraction methods auto encoder, principal component analysis, and the restricted Boltzmann machine on the network's overall ability to forecast future market movements. Empirical results suggest that deep neural networks can extract valuable features from the residuals of the autoregressive model and enhance forecasting performance. However, when the autoregressive model is applied to the residuals of the network, it can not extract the same valuable information. Chong et al. (2017) provide useful directions for further examination into how deep learning networks can be satisfactorily used for stock market forecasting.

Among the deep learning models, CNN has recently become one of the favorable financial time series trend analyzers and performs magnificent results. Although CNNs are applied to financial time series to get a better result, financial time series cannot be used directly on a convolutional neural network. Some operational processes are required. So that financial time series are transformed into the 2D-array format. CNN extracts features from the 2D array for forecasting the financial time series movements. Sezer and Ozbayoglu (2018) built 2-D images using stock technical indicators and proposed a novel algorithmic trading expert system to predict stock price trends (Sezer and Ozbayoglu, 2018).

Nelson et al., (2017) use LSTM networks to forecast future movements of financial time series based on the past values with financial technical analysis indicators. For this purpose, many experimental results were analyzed via several metrics to evaluate the algorithm when compared to other machine learning algorithms. An average of 55.9% of accuracy was obtained for the future trend of a particular stock (Nelson et al., 2017). Moews et al. (2019) focused on complex lagged correlations between forecasting of directional trend changes using deep feed-forward neural networks joined with exponential smoothing for financial time series trend prediction. Chen et al. (2019) investigated the public emotion and mood from news articles and then used the LSTM network to forecast the future trend of the stock. Liang et al. (2017) proposed to integrate RBM and several classifiers to forecast short-term stock market trends. Naik and Mohan (2020) proposed a method to determine stock movement trends using a deep neural network (DNN) on a combination of candlestick data and technical indicators. Zhao et al. (2021) proposed forecasting methods based on RNN/LSTM/GRU, respectively. The attention mechanism can choose and pay attention to “key information”. Hence, based on the conventional Recurrent Neural Network, this paper included the attention mechanism and proposed a forecasting model based on AT-RNN/AT-LSTM/AT-GRU evaluation of RNN-M, LSTM-M, and GRU-M forecasting models, the GRU-M and LSTM -M was considerably better than the RNN-M, and the GRU-M was a little better than the LSTM-M. In addition to these, CNN-LSTM models have been proposed to determine the stock trends in stock markets (Liu et al., 2017), and bidirectional LSTM was also used in financial time series analysis (Yanga and Wang 2022).

2.4. Machine Learning Approaches in BIST Studies

Borsa Istanbul (BIST) is the primary stock exchange of Turkey that follows financial news articles, financial technical indicators, and historical index prices that have grown with great interest in current years. More than 500 companies are listed on the BIST. The BIST 30, BIST 50, and BIST 100 (XU100) indexes are listed on the Istanbul Stock Exchange.

Aydin and Cavdar (2015) proposed to predict the relationship between the exchange rate of the US Dollar-Turkish Lira (USD/TRY), gold prices, and the Borsa Istanbul (BIST) 100 index using an ANN algorithm and Vector Autoregressive (VAR) model. The ANN algorithm has been utilized using multi-layered feed-forward neural networks (MLFNs) by using monthly data between 2000 and 2014 years.

Alp et al. (2020) proposed to define whether the deep learning methods outperform traditional ARIMA method in forecasting the BIST 30, BIST 50, and BIST 100 price indices. The forecasting performance of ARIMA was compared among the Long Short-Term Memory (LSTM) and Gated-Recurrent Unit (GRU) for each BIST price index. The root mean square error is used for evaluation metric. It has been showed that ARIMA methods have better performance in forecasting BIST 30, BIST 50, and BIST 100 indices than deep learning models.

Direction of movement in Borsa Istanbul (BIST) National 100 is predicted by Random Forest (RF), Support vector machine (SVM), and Linear Regression (LR) over a set of sixty-four financial technical indicators that were taken as input features to predict the oncoming (one-period-ahead) direction of BIST 100 (Hasan et al., 2018).

Pehlivanlı et al. (2016) proposed a method to predict the next day's stock price movement using the optimal subset indicators selected with an ensemble feature selection approach. The aim is to get the final best feature subset and remove redundant and irrelevant indicators from the dataset. For this objective, filter methods are joined, support vector machines (SVM) have been carried out, and finally, a voting scheme is applied. Istanbul Stock Exchange(ISE) Data is used. The result expresses that the filtered dataset predicts better results than the full dataset.

Bildirici and Ersin (2009) proposed to analyze the ARCH/GARCH family models and improve them with artificial neural networks to assess the volatility of daily returns between the 23.10.1987 and 22.02.2008 period in the Istanbul Stock Exchange. They found that the ANN-APGARCH model increased the forecasting performance of the APGARCH model.

Boyacioglu and Avci (2010) researched the future movements of stock market return with Adaptive Network-Based Fuzzy Inference System (ANFIS). The aim is to examine whether an ANFIS algorithm is able to forecast accurately stock market returns. ANFIS uses six macroeconomic variables and three indices as input variables. The experimental results show that ANFIS successfully predicts with an accuracy rate of 98.3%, the monthly return of the ISE National 100 Index.

3. MATERIALS AND METHODS

In this section of the thesis, the materials and methods used in the studies are explained in detail. We describe our proposed novel CNN-DBEMA method using a convolutional neural network with a combined distance-based exponential moving-average features and technical analysis (TA) indicators to forecast financial time series. As is shown in Fig. 3.1, the proposed CNN-DBEMA-based system follows these main phases: dataset preparation, computing technical indicators based features, forming distance-based exponential moving average (DBEMA) based features, automated labeling of the dataset, creating an image for each data instance, normalization of the data instances, and trend prediction. To briefly explain Fig.3.1, the process starts with the gathering of the raw data of stocks, which contain open, high, low, close, and volume values, from the Finnet application. After getting bulk raw data from the stocks, each stock's data are stored in Excel files. To generate financial indicators and novel distance-based indicators, the FinanceLab tool was developed using the Python programming language. In the next phase of data standardization, each stock's Excel file data is processed in FinanceLab Class and each stock's Excel file data is used to generate financial indicators-based features, novel distance-based features, and labeling with its class name. In the labeling phase, each stock's Excel file is read and assigned a class label from the set of class labels that is "earn", "earning trap", "loss", and "losing trap" according to Algorithm 1 and Algorithm 2. From 118 BIST stocks data for approximately 20 years period, 6425 labeled images for CNN operations were produced. The 6425 image files were split into two parts: Images obtained from the data of years from 1998 to 2017 were used for training the CNN, and images of years from 2017 to 2021 were used for testing operations. Then, in the evaluation phase, training and testing data sets were used in the following benchmarking models: linear regression (LR), random forests (RF), gaussian naïve bayes (NB), k nearest neighbor (KNN), multi-layer perceptron (MLP), bagged trees regressor (BAG), a novel CNN-TA, and the relative strength index method (RSI). At the end of the operations, the proposed models and benchmarking models were compared to each other.

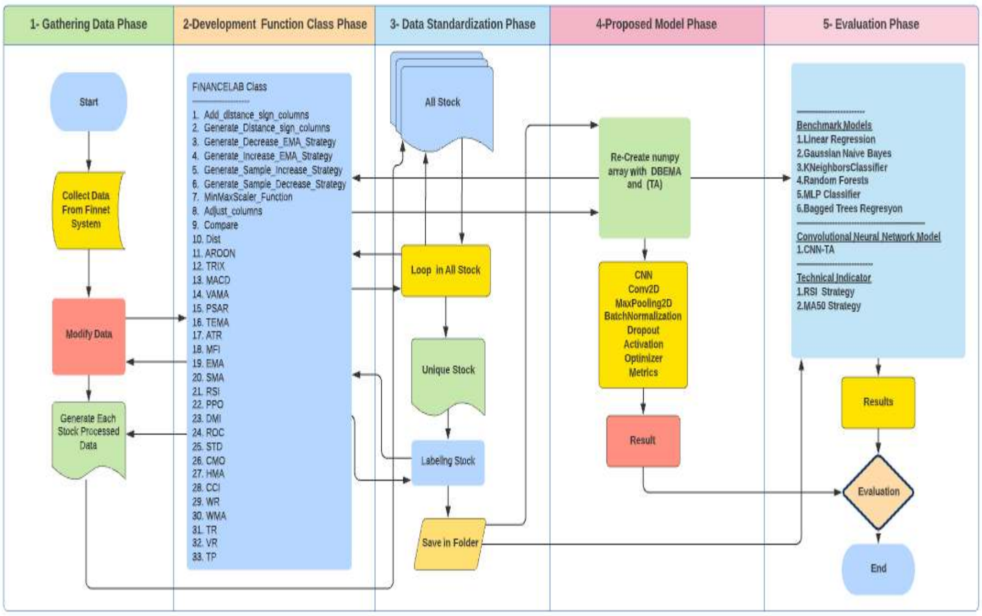


Figure 3.1. Flow diagram of the CNN-DBEMA model.

3.1. Data Description

Gathering raw data of stocks listed in BIST starts with getting daily stock prices from the Finnet application. 118 BIST stocks were used in the thesis. The stocks used in the thesis are shown in Table 3.1. Each raw stock data contains open, high, low, close, and volume values for 5701 trading days between June 1998 and April 2021. Fig. 3.2 shows us the open, high, low, and close (OHLC) prices of the sample stock GARAN, which is one of the biggest banks in the BIST 100 index. Each stock's data are stored in Excel files. The type of excel file is shown in Fig.3.3.

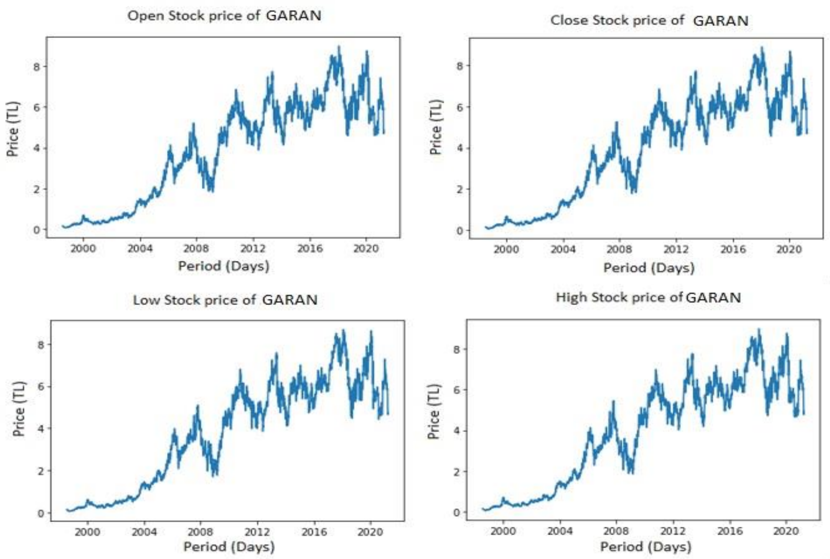
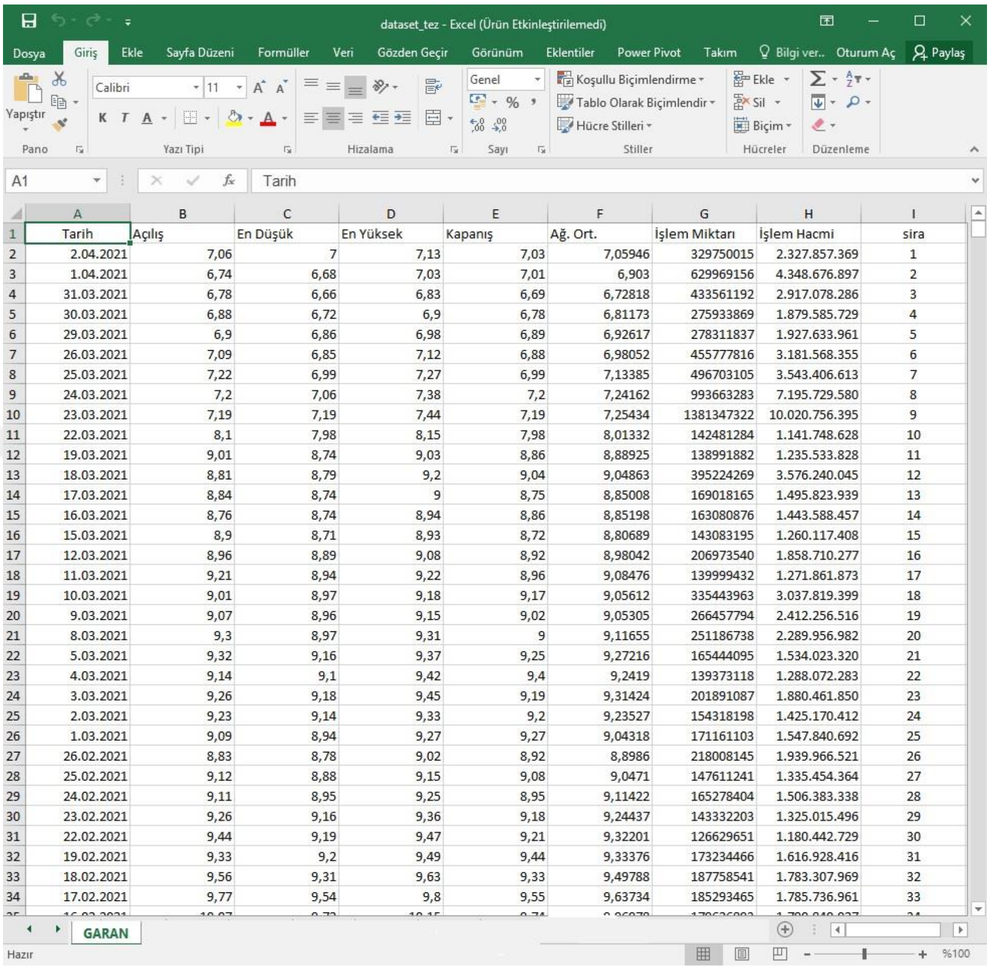


Figure 3.2. Price of GARAN stock (OHLC).



Tarih	Açılış	En Düşük	En Yüksek	Kapanış	Ağ. Ort.	İşlem Miktarı	İşlem Hacmi	sira
2.04.2021	7,06	7	7,13	7,03	7,05946	329750015	2.327.857.369	1
1.04.2021	6,74	6,68	7,03	7,01	6,903	629969156	4.348.676.897	2
31.03.2021	6,78	6,66	6,83	6,69	6,72818	433561192	2.917.078.286	3
30.03.2021	6,88	6,72	6,9	6,78	6,81173	275933869	1.879.585.729	4
29.03.2021	6,9	6,86	6,98	6,89	6,92617	278311837	1.927.633.961	5
26.03.2021	7,09	6,85	7,12	6,88	6,98052	455777816	3.181.568.355	6
25.03.2021	7,22	6,99	7,27	6,99	7,13385	496703105	3.543.406.613	7
24.03.2021	7,2	7,06	7,38	7,2	7,24162	993663283	7.195.729.580	8
23.03.2021	7,19	7,19	7,44	7,19	7,25434	1381347322	10.020.756.395	9
22.03.2021	8,1	7,98	8,15	7,98	8,01332	142481284	1.141.748.628	10
19.03.2021	9,01	8,74	9,03	8,86	8,88925	138991882	1.235.533.828	11
18.03.2021	8,81	8,79	9,2	9,04	9,04863	395224269	3.576.240.045	12
17.03.2021	8,84	8,74	9	8,75	8,85008	169018165	1.495.823.939	13
16.03.2021	8,76	8,74	8,94	8,86	8,85198	163080876	1.443.588.457	14
15.03.2021	8,9	8,71	8,93	8,72	8,80689	143083195	1.260.117.408	15
12.03.2021	8,96	8,89	9,08	8,92	8,98042	206973540	1.858.710.277	16
11.03.2021	9,21	8,94	9,22	8,96	9,08476	139999432	1.271.861.873	17
10.03.2021	9,01	8,97	9,18	9,17	9,05612	335443963	3.037.819.399	18
9.03.2021	9,07	8,96	9,15	9,02	9,05305	266457794	2.412.256.516	19
8.03.2021	9,3	8,97	9,31	9	9,11655	251186738	2.289.956.982	20
5.03.2021	9,32	9,16	9,37	9,25	9,27216	165444095	1.534.023.320	21
4.03.2021	9,14	9,1	9,42	9,4	9,2419	139373118	1.288.072.283	22
3.03.2021	9,26	9,18	9,45	9,19	9,31424	201891087	1.880.461.850	23
2.03.2021	9,23	9,14	9,33	9,2	9,23527	154318198	1.425.170.412	24
1.03.2021	9,09	8,94	9,27	9,27	9,04318	171161103	1.547.840.692	25
26.02.2021	8,83	8,78	9,02	8,92	8,8986	218008145	1.939.966.521	26
25.02.2021	9,12	8,88	9,15	9,08	9,0471	147611241	1.335.454.364	27
24.02.2021	9,11	8,95	9,25	8,95	9,11422	165278404	1.506.383.338	28
23.02.2021	9,26	9,16	9,36	9,18	9,24437	143332203	1.325.015.496	29
22.02.2021	9,44	9,19	9,47	9,21	9,32201	126629651	1.180.442.729	30
19.02.2021	9,33	9,2	9,49	9,44	9,33376	173234466	1.616.928.416	31
18.02.2021	9,56	9,31	9,63	9,33	9,49788	187758541	1.783.307.969	32
17.02.2021	9,77	9,54	9,8	9,55	9,63734	185293465	1.785.736.961	33

Figure 3.3. Representation of the excel data of GARAN stock.

The dataset was separated into two parts: the training part (from June 1998 to December 2016) and the testing part (from January 2017 to April 2021). The reason for choosing the companies in Table 3.1 is that these companies are both stable and have the most comprehensive historical data. In addition, the dataset between June 1, 1998, and April 21, 2021, covers the 2001 and 2008 financial crisis periods.

Table 3.1. The list of BIST companies used in this thesis.

No	Stock Code	No	Stock Code	No	Stock Code	No	Stock Code
1	ADEL	31	CIMSA	61	GOODY	91	NTHOL
2	AKBNK	32	CLEBI	62	GOLTS	92	OLMIP
3	AKCNS	33	CMBTN	63	GUBRF	93	OYAKC
4	AKGRT	34	CMENT	64	HEKTS	94	OZGYO
5	AKSGY	35	CRDFA	65	ICBCT	95	PEGYO
6	ALARK	36	DAGHL	66	HURGZ	96	PNSUT
7	ALCAR	37	DERIM	67	IHLGM	97	RAYSG
8	ALCTL	38	DEVA	68	INTEM	98	SKBNK
9	ARCLK	39	DGGYO	69	ISCTR	99	SKTAS
10	ARSAN	40	DGKLB	70	ISYAT	101	SONME
11	ASELS	41	DITAS	71	IZMDC	101	SNPAM
12	ASUZU	42	DOGUB	72	KAPLM	102	TATGD
13	ATAGY	43	DYOBY	73	KARTN	103	TIRE
14	ATEKS	44	DZGYO	74	KCHOL	104	TRCAS
15	ATLAS	45	EGEEN	75	KENT	105	TOASO
16	AVGYO	46	EGGUB	76	KERVN	106	TSKB
17	AYGAZ	47	EGPRO	77	KERVT	107	TUKAS
18	BAKAB	48	EGPRO2	78	KLMSN	108	ULKER
19	BANVT	49	EMKEL	79	KLNMA	109	USAK
20	BFREN	50	ENKAI	80	KNFRT	110	VAKKO
21	BOSSA	51	ERBOS	81	KONYA	111	VAKFN
22	BRISA	52	EREGL	82	KRDMB	112	VESTL
23	BRMEN	53	FMIZP	83	KRDMD	113	VKFYO
24	BRSAN	54	FRIGO	84	KRSTL	114	VKGYO
25	BRYAT	55	FROTO	85	KRTEK	115	VKING
26	BTCIM	56	GARAN	86	KUTPO	116	YATAS
27	BUCIM	57	GARFA	87	MAKTK	117	YKBNK
28	BURCE	58	GENTS	88	MGROS	118	YYAPI
29	CELHA	59	GLYHO	89	METUR		
30	CEMTS	60	GRNYO	90	NETAS		

3.2. Generating Features

Indicators are mathematical calculation methods used to analyze financial instrument data. In the literature, it is seen that technical analysis indicators are used as inputs in machine learning (Dasha and KishoreDash, 2016). Features of technical analysis are extracted from stock prices. Indicators help researchers to determine the future trend of financial time series by giving signals about financial instruments. Even though the signals give us information about the trend of the financial time series, it would not be robust to process with a single indicator. In recent years, machine learning models have been used to determine this nonlinear and complex relationship between the indicators (Nassirtoussi et al., 2014).

3.2.1. Generating Technical Indicator Features

With raw data, 37 different technical indicators were generated as input features and 15 out of 37 technical indicators were used in Pehlivanlı et al. study (2016). *MFI_EMA_9, RSI_EMA_9, RSI_MA_14, MFI_MA_3, ROC_9* and DBEMA features are designed by our study. In Table 3.2, technical indicators are listed (Pehlivanlı et al., 2016; Oriani and Coelho, 2017; Dey and Nahar, 2020). Firstly, we take the open price, high price, low price, and close price values of stocks from the excel file. According to the below expressions, the technical indicator features are listed in Table 3.2 are generated from those stock prices.

Table 3.2. Technical Indicators Used To Extract Features

No	Indicator (Input)	Description
1	SMA	Simple Moving Average
2	EMA	Exponential Moving Average
3	RSI	Relative Strength Index
4	MFI	Money Flow Index
5	ROC	Rate of Change
6	MFI_EMA_9	9-days EMA of MFI
7	RSI_EMA_9	9-days EMA of RSI
8	TR	True Range
9	ATR	Average True Range
10	DI+	Positive Directional Indicator
11	DI-	Negative Directional Indicator
12	DMI	Directional Movement Index
13	CMFI	Chaikin Money Flow Index
14	TP	Typical Price
15	TRIX	Triple exponential Average
16	MACD	Moving Average Convergence Divergence
17	VAMA	Volume Adjusted Moving Average
18	PSAR	Parabolic Stop and Reverse
19	TEMA	Triple Exponential Moving Average
20	PPO	Percentage Price Oscillator
21	CMO	Chande Momentum Oscillator
22	WMA	Weighted Moving Average
23	HMA	Hull Moving Average
24	CCI	Commodity Channel Index
25	WR	Larry William %R
26	RSI_MA_14	14-days MA of RSI
27	MFI_MA_3	3-days EMA of MFI
28	PPO_histo	PPO- PPO_histo
29	PPO_signal	9-days WMA of PPO
30	MACD_signal	9-days WMA of MACD
31	ROC_9	9-days EMA of ROC
32	EMA9_EMA50_dist	Distance Between EMA9 and EMA50
33	EMA9_EMA200_dist	Distance Between EMA9 and EMA200
34	EMA50_EMA200_dist	Distance Between EMA50 and EMA200
35	EMA9_EMA50_dist_sign	Comparison of EMA9 and EMA50
36	EMA9_EMA200_dist_sign	Comparison of EMA9 and EMA200
37	EMA50_EMA200_dist_sign	Comparison of EMA50 and EMA200

Simple Moving Average (SMA): A simple moving average (SMA) is the average of a stock's price over a period of time. SMA is generally used to determine trend direction. In this thesis 14-days (Aycel and Santur 2022) period is used to calculate the SMA and the formula is denoted in the below expression, where $\mathrm{Price_t}$ is the closing price value for each time unit in the relevant period and *n* is the size of the period.

$$SMA = \sum_{t=1}^{n} \mathrm{Price_t} \qquad (3.1)$$

Exponential Moving Average (EMA): Exponential Moving Average (EMA) is a type of moving average (MA) that puts more emphasis on recent data points. The reason for using exponential moving averages is that the exponential moving average (EMA) responds rapidly when the price of stocks changes in the financial market. In this thesis 9-days, 50-days, and 200-days periods are used to calculate the EMA and the formula is denoted in the below expression.

$$EMA = \mathrm{Price_t} \text{ x } \mathrm{multiplier} + \mathrm{EMA}\ (\mathrm{Price_{t-1}}) \text{ x } (1 - \mathrm{multiplier}) \qquad (3.2)$$

where the multiplier is computed as follows:

$\mathrm{multiplier} = \left(\frac{2}{(\text{number of observations} + 1)}\right)$ in which number of observations is the total number of closing values that are contained in the EMA.

Relative strength index (RSI): The relative strength index (RSI) is a momentum indicator used in technical analysis that measures the significance of recent price changes to assess oversold or overbought positions in the price of a stock. It measures a number from 0 to 100. Using this measurement, buy-sell signals are produced for the researcher. The most common strategy in the market is to buy when a 14-day period RSI falls below 30 and sell when it rises above

70 (Chen et al., 2021). The default RSI setting for the RSI indicator is 14-periods. The Average Gain input of the close stock price is higher than the previous price. The Average Loss input of the stock close price is lower than the previous price. RSI is computed according to equation 3.3, and Average Gain and Average Loss are computed according to the below algorithm:

$$RSI = 100 - \frac{100}{1+(Average\ Gain/Average\ Loss)} \tag{3.3}$$

Algorithm: Compute Average Gain and Average Loss

Input: return $= (\text{Close Price}_t\text{-Close Price}_{t-1})/\text{Close Price}_{t-1}$

Output: Average Gain and Average Loss

Set initial values total gain = 0, total loss = 0

Period = 14-days (value = 14)

for t =1 **to** Period:

  if (return) > 1 :

    total gain = total gain + |return|

  else:

    total loss = total loss + |return|

end for

Average Gain = total gain / Period

Average Loss = total loss / Period

Money Flow Index (MFI): The Money Flow index (MFI) is a momentum indicator used in technical analysis that measures the significance of recent price changes to assess. When MFI gets over 80 it is considered overbought and when MFI moves down below 20 it is considered oversold.

$$\text{MFI} = 100 - \frac{100}{1+(\text{Monef Flow Ratio})} \tag{3.4}$$

$$\text{Money Flow Ratio} = \frac{\text{Positive Money Flow(14\_days\_Period)}}{\text{Negative Money Flow(14\_days\_Period)}}$$

where money flow ratio is computed as follows:

$$\text{Raw Money Flow} = \text{Typical Price} * \text{Volume}$$

$$\text{Typical Price} = \frac{\text{High}_t + \text{Low}_t + \text{Close}_t}{3}$$

In the Positive Money Flow phase, the Raw Money Flow increase from one period to the next period if Raw Money Flow is positive and it is added to Positive Money Flow.

In the Negative Money Raw phase, the Raw Money Flow decreases from one period to the next period. If Raw Money Flow is negative and it is added to Negative Money Flow.

Rate of Change (ROC): Rate of change is used to define the percentage change in stock prices over a defined period, and it symbolizes the momentum of a parameter. The calculation of the rate of change is denoted in the below expression.

$$ROC = 100 * \left(\frac{\text{Price}_t}{\text{Price}_{t-1}} - 1\right) \tag{3.5}$$

9 days EMA of MFI (MFI_EMA_9): The MFI_EMA_9 is an indicator that is more robust than MFI. It is defined by our study. The reason for the selected nine days EMA of MFI is ema responses more quickly, and 9 days is generally used in the finance literature (Pardeshi and Kale, 2021). Thus, we will have made a more stable

MFI approach. The calculation of 9 days EMA of MFI is denoted in the below expression.

$$MFI\_EMA\_9 = \mathrm{EMA}(\mathrm{MFI}, 9) \tag{3.6}$$

9 days EMA of RSI (RSI_EMA_9): The RSI_EMA_9 is an indicator that is more robust than RSI. It is defined by our study. The reason of the selected 9 days EMA of RSI is ema responds more quickly, and 9 days is generally used in the finance literature (Pardeshi and Kale, 2021). Thus, we will have a more stable RSI approach. The calculation of 9 days EMA of RSI is denoted in the below expression.

$$RSI\_EMA\_9 = \mathrm{EMA}(\mathrm{RSI}, 9) \tag{3.7}$$

True Range (TR): The True range is a measurement of volatility. It displays how many financial instruments change the price on a given day. TR is computed as below:

$$\mathrm{TR} = \max[(\mathrm{high} - \mathrm{low}), \mathrm{abs}(\mathrm{high} - \mathrm{close}_{t-1}), \mathrm{abs}(\mathrm{low} - \mathrm{close}_{t-1})] \tag{3.8}$$

where high means the highest value of the asset for the current period low is the lowest value of the asset for the current period, and $\mathrm{close}_{\mathrm{previous}}$ is the closing value of the asset for the previous period.

Average True Range (ATR): The average true range (ATR) is a volatility indicator that displays how much financial instruments move, on average, during a given period. In this thesis, a 14-days period is used for the average true range (Chang et al., 2019). It is computed as in Eq. 3.9.

$$\mathrm{ATR} = \sum_{i=1}^{n} \mathrm{TR}_i \tag{3.9}$$

Positive Directional Indicator (+DI): The Positive Directional Indicator (+DI) is used to calculate the presence of an uptrend. When the +DI is rising upward, it is a signal that the uptrend is getting stronger. In this thesis, the 14-days period is used for +DI (Ntakaris, A., et al. 2020). In Eq. 3.11, it's formula is given:

$$+\mathrm{DI} = \left(\frac{\mathrm{S{+}DM}}{\mathrm{ATR}}\right) * 100 \tag{3.10}$$

Negative Directional Indicator (-DI): The Negative Directional Indicator (-DI) is used to calculate the presence of a downtrend. When the -DI is rising upward, it is a signal that the downtrend is getting stronger. In this thesis, the 14-days period is used for -DI (Ntakaris et al., 2020). In Eq. 3.11, It is computed as follows:

$$-\mathrm{DI} = \left(\frac{\mathrm{S{-}DM}}{\mathrm{ATR}}\right) * 100 \tag{3.11}$$

where

$\text{-DM} = \text{Previous Low - Current Low}$

$+\text{DM} = \text{Current High - Previous Low High}$

$\text{S-DM} = \sum_{t=t-14}^{t} -\mathrm{DM} + \left(\frac{\sum_{t=t-14}^{t} -\mathrm{DM}}{14}\right) + \text{Current} - \mathrm{DM}$

$\text{S+DM} = \sum_{t=t-14}^{t} +\mathrm{DM} + \left(\frac{\sum_{t=t-14}^{t} +\mathrm{DM}}{14}\right) + \text{Current} + \mathrm{DM}$

Directional Movement Index (DMI): The Directional Movement Index (DMI) is an indicator that defines the direction the financial time series is moving and the strength of its movement. In this thesis, the 14-day period is used for DMI (Ntakaris et al., 2020). It is computed as in Eq. 3.12.

$$\mathrm{DMI} = \left(\frac{|(+\mathrm{DI})-(-\mathrm{DI})|}{|(+\mathrm{DI})+(-\mathrm{DI})|}\right) * 100 \tag{3.12}$$

Chaikin Money Flow Index (CMFI): Chaikin Money Flow Index is a volume-weighted average of aggregation and delivery over a specified period.

In Chaikin Money Flow the nearer the closing price is to the high, the more aggregation has taken place. On the other hand, the nearer the closing price is to the low, the more delivery has taken place. If the price action consistently closes above the bar's midpoint on increasing volume, the Chaikin Money Flow will be positive. On the other hand, if the price action consistently closes below the bar's midpoint on increasing volume, the Chaikin Money Flow will be a negative value. In this thesis, the 14-days period is used for CMFI (Jiang and Chang, 2020.). It is computed as in Eq. 3.13.

CMFI = 21-day Average of the Daily Money Flow / 21-day Average of the Volume (3.13)

where MFM is Money Flow Multiplier, and it is computed as follows

MFM = ((Close– Low) – (High– Close)) / (High– Low)

In the above equation

High is current period highest value of assets,

Low is current period lowest value of assets,

Close is current period closed value of asset.

By using MFM, money flow volume is computed as below:

MFV = MFM x Volume for the Period

Typical Price (TP): The Typical Price indicator takes the simple average between the high, low, and closing prices. The Typical Price is denoted by the below expression.

$$\text{Typical Price} = \frac{\text{High+Low+Close}}{3} \tag{3.14}$$

Triple Exponential Average (TRIX): The triple exponential average (TRIX) is a momentum indicator that indicates the percentage change in a moving average that has been smoothed exponentially three times.

The triple smoothing of moving averages is devised to screen price movements that are considered unimportant. It is computed as in Eq. 3.15:

$$\mathrm{TRIX}_{(i)} = \frac{\mathrm{EMA3}_{(i)} - \mathrm{EMA3}_{(i-1)}}{\mathrm{EMA3}_{(i-1)}} \tag{3.15}$$

where

$\mathrm{EMA3}_{(i)} = \mathrm{EMA}(\mathrm{EMA2}_{(i)}, \mathrm{N})$

$\mathrm{EMA2}_{(i)} = \mathrm{EMA}(\mathrm{EMA1}_{(i)}, \mathrm{N})$,

$\mathrm{EMA1}_{(i)} = \mathrm{EMA}(\mathrm{Price}, \mathrm{N})$ which is the current value of the Exponential Moving Average, and $\mathrm{Price}_{(i)}$ is the currently closed value of the asset.

Moving Average Convergence/Divergence (MACD): The Moving Average Convergence/Divergence indicator is a momentum indicator that shows the relationship between two different moving averages of prices. It is fundamentally produced by subtracting the 26-day EMA from the 12-day EMA. The crossover of the two EMA lines gives trading signals similar to a two-moving average system(Lei et al., 2020). The calculation of MACD is denoted by the below expression in Eq 3.16.

$$\mathrm{MACD} = \mathrm{EMA}(\text{closed value}, 12) - \mathrm{EMA}(\text{closed value}, 26) \tag{3.16}$$

Volatility Adaptive Moving Average (VAMA): The Volatility Adaptive Moving Average indicator gives an adjusted moving average, based on the volatility of the past *N* amount of periods, measured against the EMA of a certain length. The calculation of VAMA is denoted by the below expression in Eq 3.17.

$$\mathrm{VAMA} = \frac{\sum_{i=0}^{N}(\mathrm{Price}_i * \mathrm{Volume}_i)}{\sum_{i=0}^{N}(\mathrm{Volume}_i)} \tag{3.17}$$

Parabolic Stop and Reverse (PSAR): The Parabolic SAR is a technical indicator that identifies the financial time series direction. The indicator is also described as a stop and reverse system. It aims to determine potential reversals in the price movement of financial time series. The calculation of PSAR is shown in the below expression in Eq 3.18.

$$\mathrm{PSAR} = \mathrm{Prior\ PSAR} + \mathrm{Prior\ AF}\ (\mathrm{Prior\ EP} - \mathrm{Prior\ PSAR}) \tag{3.18}$$

In Eq. 3.18, EP is the highest high for an uptrend and lowest low for a downtrend updated each time a new EP is reached. AF is equal to 0.02 by default, and it is increased by 0.02 each time a new EP is reached, with a maximum of 0.20.

Triple Exponential Moving Average (TEMA): The triple exponential moving average (TEMA) was designed to flat financial time series changing,
therefore making it easier to determine trends without the lag associated with traditional moving averages (MA). It achieves this by taking multiple exponential moving averages (EMA) of the original EMA and subtracting out some of the lag. The calculation of TEMA is shown in the below formula in Eq 3.19.

$$\mathrm{TEMA} = (3 * \mathrm{EMA1}) - (3 * \mathrm{EMA2}) + (\mathrm{EMA3}) \tag{3.19}$$

Percentage Price Oscillator (PPO): The PPO indicator is based on a moving average that supports the researcher in discovering in what direction a particular financial time series may move in the near future. It uses two moving averages – one over a shorter period and one over a longer period. It averages that information to display a trend. The calculation of PPO is shown in the below formula in Eq 3.20.

$$\mathrm{PPO} = \frac{\mathrm{EMA(12\ Period)} - \mathrm{EMA(26\ Period)}}{\mathrm{EMA(26\ Period)}} \tag{3.20}$$

Chande Momentum Oscillator (CMO): The Chande Momentum Oscillator's calculations are just the total net change over a period divided by the absolute value of the individual net changes over that period. The values of the CMO intervals are between – 100 and + 100. When CMO values are above 50, it means overbought conditions conditions for the financial instrument, while the CMO values are below – 50 mean financial instrument is oversold. In this thesis, the system period is set as 9 days (Wang, 2020). The calculation of CMO is shown in the below formula in Eq 3.21.

$$\mathrm{CMO} = \frac{\mathrm{sumHigh} - \mathrm{sumLow}}{\mathrm{sumHigh} + \mathrm{sumLow}} \tag{3.21}$$

where sumHigh is the sum of the difference between today's close and yesterday's over N periods, and sumLow displays the absolute value of the difference between today's close price and yesterday's close price on down days.

Weighted Moving Average (WMA): The weighted moving average (WMA) is a technical indicator that allocates a greater weighting to the most recent financial instrument values and less weighting to financial instrument values in the distant past. The WMA is found by multiplying each number in the data set by a predefined weight and summing up the resulting values. In this thesis, the system period is set as a 9-days period(Sezer and Ozbayoglu, 2018). The calculation of WMA is shown in the below formula in Eq 3.22.

$$\mathrm{WMA} = \frac{(\mathrm{Closed\ Value}_1 * n) + (\mathrm{Closed\ Value}_2 * n-1) + \cdots + \mathrm{Closed\ Value}_n}{[n*(n+1)]/2} \tag{3.22}$$

where n is the time period count.

Hull Moving Average (HMA): The Hull Moving Average (HMA) efforts to reduce the lag of a traditional moving average while maintaining the flatness of the moving average line. This indicator is based on weighted moving averages to emphasize more recent values for reducing lag. The resulting averaging is more sensitive and well-suited for determining entry points. . In this thesis, the system period is set as a 9-days period (O.B. Sezer, and A.M. Ozbayoglu, 2018). The calculation of WMA is shown in the below formula in Eq 3.23.

$$\mathrm{HMA} = \mathrm{WMA}(2 * \mathrm{WMA}(n/2) - \mathrm{WMA}(n)), \mathrm{sqrt}(n)) \qquad (3.23)$$

Commodity Channel Index (CCI): The Commodity Channel Index (CCI) is a momentum-based indicator used to help decide when a researcher's financial instrument is reaching a status of being overbought or oversold. CCI Indicator makes movements within the reference value range. Above +100 is considered overbought, and below -100 is considered oversold (Wang, S., 2020). The calculation of CCI is shown in the below formula in Eq 3.24.

$$\mathrm{CCI} = (\mathrm{TP} - \text{14\_period SMA of TP}) \,/\, (0.015 * \mathrm{Dev}) \qquad (3.24)$$

where Dev is computed as below, TP is shown in In Eq. 3.14 and SMA is shown in Eq. 3.1, and n is the period. $\mathrm{Dev} = \frac{\sum_{t=1}^{n} \mathrm{TP}-\mathrm{SMA}}{n}$

Larry William %R (WR): The Williams %R is a momentum indicator that is the opposite of the Fast Stochastic indicator. When Williams %R takes values between 0 and -20, it is considered overbought. On the otherhand, Williams %R takes values between -80 to -100 are considered oversold. Williams %R considers the level of the close relative to the highest high for the look-back period(Wang, 2020). The calculation of WR is shown in the below formula in Eq 3.25.

$$\text{WR} = \frac{(\text{Highest High value} - \text{Close value})}{(\text{Highest High value} - \text{Lowest Low value})} * -100 \tag{3.25}$$

14-days MA of RSI (RSI_MA_14): The RSI_MA_14 is an indicator that is more robust than RSI. It is defined by our study. The reason for the selected 14 days SMA of RSI is sma responds more slowly, and 14 is generally used in finance literature. Thus, we will have made a more stable RSI approach. The calculation of 14 days SMA of RSI is denoted below expression.

$$RSI\_MA\_14 = \text{SMA}(\text{RSI}, 14) \tag{3.26}$$

3-days EMA of MFI (MFI_MA_3): The MFI_MA_3 is an indicator that is more robust than MFI. It is defined by our study. The reason for the selected 3 days EMA of MFI is ema responses more quickly, and 3 is short term to observed shortchanging in MFI. Thus, we will have made more quick changes in MFI. The calculation of 3 days EMA of MFI is denoted in below expression.

$$MFI\_MA\_3 = \text{EMA}(\text{MFI}, 3) \tag{3.27}$$

PPO- PPO_histo (PPO_Histo): The percentage price oscillator (PPO) is a technical momentum indicator that denotes the relationship between two exponential moving averages. The moving averages are a 26-days period and a 12-days period exponential moving average (EMA). The PPO is used to compare financial instrument performance and volatility that could manage to price turnaround, produce trade signals, and assist verify trend direction(Wang, 2020). The calculation of PPO_histo is shown in below expression.

$$\mathrm{PPO} = \frac{\mathrm{EMA(12\ Period)} - \mathrm{EMA(26\ Period)}}{\mathrm{EMA(26\ Period)}}$$

$$\mathrm{PPO_{Histo}} = \mathrm{PPO} - \mathrm{PPO}\ \mathit{Signal\ Line} \tag{3.28}$$

$$\mathrm{PPO}\ \mathit{Signal\ Line} = \mathrm{EMA(PPO\,,9)} \tag{3.29}$$

9-days EMA of MACD (MACD_signal): A 9-day EMA of the MACD called the MACD_signal, which can operate as a trigger for buy and sell signals(Lei, Y., et al. 2020). When the MACD crosses above its signal line and it means the financial time series goes uptrend when the MACD crosses below the signal line, it means the financial time series goes downtrend.

$$MACD\_signal = \mathrm{EMA(MACD\,,9)} \tag{3.30}$$

9-days EMA of ROC (ROC_9): The *ROC_9* is an indicator that supports the researcher to make robust decisions. It is defined by our study. The reason for the selected 9 days EMA of ROC is EMA responses more rapidly, and 9 days is generally used in the finance literature. Thus, we will have made a more stable ROC approach. The calculation of 9 days EMA of ROC is denoted below expression.

$$\mathrm{ROC\_9} = \mathrm{EMA\ (ROC\,,9)} \tag{3.31}$$

3.2.2. Generating A Novel Distance-Based Features

Distance Between EMA9 and EMA50 (EMA9_EMA50_dist): The EMA9_EMA50_dist is the novel approach to predicting the relationship between financial time series and its trend. The strength of distance gives how strongly the time series will change its position. In this thesis, EMA9 is the fastest exponential moving average and EMA50 is the fast exponential moving average. Fig 3.4 presents distances between EMA9 and EMA50. The calculation of distance between EMA9 and EMA50 is denoted in the below expression.

$$\text{EMA9\_EMA50\_dist} = \text{abs}\left(\frac{\text{Ema9} - \text{Ema50}}{\text{Ema9}}\right) \tag{3.32}$$

Distance Between EMA9 and EMA200 (EMA9_EMA200_dist): The EMA9_EMA200_dist is the novel approach to predicting the relationship between financial time series and its trend. The strength of distance gives how strongly the time series will change its position. In this thesis EMA9 is the fastest exponential moving average and EMA200 is the slow exponential moving average. The distances between EMA9 and EMA200 are the main indicator that shows which trend of the time series will be. Fig 3.4 presents distances between EMA9 and EMA50. The calculation of distance between EMA9 and EMA200 is denoted in the below expression.

$$\text{EMA9\_EMA200\_dist} = \text{abs}\left(\frac{\text{EMA9} - \text{EMA200}}{\text{EMA9}}\right) \tag{3.33}$$

Distance Between EMA50 and EMA200 (EMA50_EMA200_dist): The EMA50_EMA200_dist is the novel approach to predicting the relationship between financial time series and its trend. The strength of distance gives how strongly the time series will change its position. In this thesis, EMA50 is the fast exponential moving average and EMA200 is the slow exponential moving average. Fig 3.4 presents distances between EMA50 and EMA200. The calculation of distance between EMA50 and EMA200 is denoted in the below expression.

$$\text{EMA50\_EMA200\_dist} = \text{abs}\left(\frac{\text{EMA50} - \text{EMA200}}{\text{EMA50}}\right) \tag{3.34}$$

Figure 3.4. The distances between moving averages

Comparison of EMA9 and EMA50 (EMA9_EMA50_dist_sign): The EMA9_EMA50_dist is the novel approach to illustrating financial time series trends. The value of the distance sign gives the time series trend. In this thesis, EMA9_EMA50_dist_sign takes two positive parameters. Positivity is a vital situation for CNN architecture. In CNN architecture inputs must take only positive values. The EMA9_EMA50_dist_sign takes 1 and 2 values. if the system set “1” value on this EMA9_EMA50_dist_sign parameter, it means this downtrend but it does not mean the time series is in a down trend. The main trend is the partial trend. if the system set “2” value on this EMA9_EMA50_dist_sign parameter, it means this is a partial uptrend in the main trend. The calculation of the distance sign between EMA9 and EMA50 is shown in the below expression.

$$\text{EMA9\_EMA50\_dist\_sign} \quad = \begin{cases} 1, & \text{if EMA9} < \text{EMA50} \\ 2, & \text{otherwise} \end{cases} \tag{3.35}$$

Comparison of EMA9 and EMA200 (EMA9_EMA200_dist_sign): The EMA9_EMA200_dist_sign is the novel approach to illustrating financial time series trends. The value of the distance sign gives the time series trend. In this thesis, EMA9_EMA200_dist_sign takes two positive parameters. Positivity is a vital situation for CNN architecture. In CNN architecture inputs must take only positive values. The EMA9_EMA200_dist_sign takes 1 and 2 values. If the system set "1" value on this EMA9_EMA200_dist_sign parameter, it means this is downtrend. If the system set "2" value on this EMA9_EMA200_dist_sign parameter, it means an uptrend. The calculation of the distance sign between EMA9 and EMA200 is denoted in the below expression.

$$\text{EMA9\_EMA200\_dist\_sign} = \begin{cases} 1, & \text{if EMA9} < \text{EMA200} \\ 2, & \text{otherwise} \end{cases} \tag{3.36}$$

Comparison of EMA50 and EMA200 (EMA50_EMA200_dist_sign): The EMA50_EMA200_dist_sign is the novel approach to illustrating financial time series trends. The value of the distance sign gives the time series trend. In this thesis, EMA50_EMA200_dist_sign takes two positive parameters. Positivity is a vital situation for CNN architecture. In CNN architecture inputs must take only positive values. The EMA50_EMA200_dist_sign takes 1 and 2 values. The sign between the distance of EMA50 and EMA200 shows the strength of financial time series trends. The calculation of the distance sign between EMA50 and EMA200 is denoted in the below expression.

$$\text{EMA50\_EMA200\_dist\_sign} = \begin{cases} 1, & \text{if EMA50} < \text{EMA200} \\ 2, & \text{otherwise} \end{cases} \tag{3.37}$$

Distance signs between EMA9, EMA50, and EMA200 values are shown in in Fig. 3.5.

Figure 3.5. The distances sign between moving averages

3.3. Design Infrastructure of the Proposed Method

Moving averages are the important technical indicators for financial time series. They perform a fundamental role in most investment strategies. They include a certain amount of data over a predefined period. In this section, we accept that EMA200 is a slow (long-term) moving average, EMA50 is medium, and EMA9 is fast (short-term). We define our system and explain how to develop the proposed exponential moving-average strategy on our system. The reason for selecting the exponential moving average is that it reacts more quickly by allocating weights to the data.

The proposed model makes its decision according to the location of exponential moving averages to each other. The proposed model is devised to forecast financial time series that produce a certain return or loss over a certain period. As we point out above, EMA crossovers are also used to define price trends. The purpose when using indicators is to determine financial time series trends. When a faster EMA9 crosses above a slow EMA200, it implies the financial time series is in an uptrend. On the other hand, a faster EMA9 crosses below a slow EMA200, it implies the financial time series is in a downtrend. According to these approaches, we defined four cases (class labels) that we called "earning class", "earning trap class", "losing class", and "losing trap class" as shown in Fig. 3.6. If the system observes an earning class, a buy signal can be offered. For losing class, a sell signal can be offered. On the other hand, for the remaining classes hold signal may be asserted as we explain in the next subsections.

Figure 3.6. Exhibition of the buy/sell signal field.

EMA crossovers find out the future movement of financial time series. On the other hand, there are scenarios that will break this approach. In order to improve the forecasting accuracy of our system, our proposed model considers 4 different cases. We demonstrate the logic of our new approach in Fig. 3.6 where the red line symbolizes EMA200, the blue line symbolizes EMA50, and the green line symbolizes EMA9. In this method, the system tries to forecast four types of classes:

i) earning class (EC), ii) earning trap class (ETC), iii) losing class (LC), and iv) losing trap class (LTC) (Ihab et al., 2009).

The system performs as follows: when a short-term EMA9 crosses above a long-term EMA200, it implies the system is creating a buy signal, but sometimes this situation can be a trap. Alternately, when a short-term EMA9 crosses below a long-term EMA200, it implies the system is creating a sell signal, and this situation can also sometimes be a trap. To overcome the trap problem, a new system, which takes a distance-based approach, was devised. When a short-term EMA9 crosses above a long-term EMA200, two possibilities exhibit themselves. One of them is "earning class", and the other is "earning trap class". In this case, the system has paid attention to both technical indicators and, our proposed features that are calculated from the distance between the exponential moving averages.

At the same time, when a short-term EMA9 crosses below a long-term EMA200, the system has paid attention to both technical indicators and our proposed features again make more efficient decisions. This novel method is illustrated in Fig. 3.7. which displays the sign and distance relations between moving averages; for example, *EMA9_EMA50_dist* means the distance between EMA9 and EMA50, *EMA9_EMA200_dist* is the distance between EMA9 and EMA200, and *EMA50_EMA200_dist* means the distance between EMA50 and EMA200. On the other hand, EMA9_EMA50_dist_sign compares the results of EMA9 and EMA50 if the EMA9 value is lower then, its value is equal to 1 otherwise equals 2 and the same convention applies to the *EMA9_EMA200_dist_sign* and *EMA50_EMA200_dist_sign* values. In this model, the distance between exponential moving averages and their position relative to each other is very significant. In equations (3.32), (3.33), (3.34), (3.35), (3.36), and (3.37), distances and their position relative to each other are illustrated. In the next subsection, we will explain how to use these features to label time series data for a given period.

Figure 3.7. Exhibition of a distance-based approach.

3.4 Labeling and Normalization Process

In this study, one of our aims is to determine how to label the financial time series data to train CNN model. The most important part is to predict the reversal point of the financial time series over a given period. Raw data labeling in time series to be used in machine learning processes is a complicated task. In our proposed model, data labeling is completely parametric. The labeling process involves two parts. In the first part, the end-of-day closing data is received from the raw data. Then, threshold values are determined for a specific time interval and specific earn, loss, earn trap, and loss trap values. As we described in section 3.3, there are four classes, defined as earning class, earning trap class, losing class, and losing trap class. In the first case when EMA9 crosses above EMA200 and the system makes at least x% profit within t days, where x is earn_ratio and t is the with_in_day parameter in Algorithm 1, this indicates an earning class and it is shown in Fig 3.8. The values of x and t are determined by the system user, and in Algorithm 1, x is chosen as 10%, and t is set to 60 days.

Figure 3.8. Presentation of an Earning Class.

In the first case when EMA9 crosses above EMA200 and the system makes at least x% profit (e.g., 10%) within t days (e.g., 60 days in Algorithm 1) this indicates an earning class and it is shown in Fig 3.8

In the second case, when EMA9 crosses above EMA200 and the system doesn't make at least x% profit—where x is earn_ratio in Algorithm 1 (e.g., $x = 10$), within t days, where t is the with_in_day parameter in Algorithm 1—this indicates an earning trap class and this case is shown in Fig 3.9

.

Figure 3.9. Presentation of an Earning Trap Class.

In the third case EMA9 crosses below EMA200 and the system produces at least *y%* loss—where *y* is loss_ratio in Algorithm 2, within *t* days, where *t* is the with_in_day parameter in Algorithm 2—this indicates a losing class. In that case, the system warns the researcher of at least the *y%*-loss ratio. At the end of the pre-defined time interval, financial instrument loss is more than the predefined *y%* loss ratio. This indicates a losing class and it is shown in Fig 3.10

In the fourth case, EMA9 crosses below EMA200 and the system doesn't produce at least *y%* loss within *t* days indicating a losing trap class. In that case, the system warns the researcher of a trap situation where *y* is loss_ratio in Algorithm 2, within *t* days, where *t* is the with_in_day parameter in Algorithm 2. At the end of the pre-defined time interval financial instrument does not lose more than the pre-defined *y%* loss ratio. This indicates a losing trap class and it is shown in Fig 3.11

Figure 3.10. Presentation of a Losing Class.

Figure 3.11. Presentation of a Losing Trap Class.

The x% profit or y% loss within t days are configurable values, and they are set by the user of the system. As a result, the proposed model is adaptable, and it can be set for any user-defined profit/loss ratio for a user-defined period of time.

The infrastructure of the labeling process is illustrated in Algorithm 1 and Algorithm 2. The labeling process indicates the movements of the stock prices after the EMA values cross each other, and the desired profit or loss amounts can be defined by the designed algorithms. The threshold points are used in determining the class label. Class labels are determined by Algorithms 1 and 2, and they are assigned to their relative 2D image-like data arrays.

After the labeling operation, we had 6425 instances and 24 attributes for each stock. This means each vector had 24 features that needed to be applied to normalization operations on each vector. In Eq. (3.38) and (3.39), a min-max transformation of each vector that was used for each stock feature is shown.

$$delta_{(i)} = \frac{size}{maxValue_{(i)} - minValue_{(i)}} \tag{3.38}$$

$$new\_value_{(i)} = (old\_value_{(i)} - minValue_{(i)}) * delta_{(i)} \tag{3.39}$$

In Eq. (3.38), size is defined as a constant value that is set randomly. But its value is initialized as 255. The new interval is between 0 and 255 because we tried to imitate the image pixel values. Delta is an equal step between intervals. The new value is set with Eq. (3.39).

Algorithm 1: Labeling Earning Class

```
Set with_in_day to 60                    // Max 60 days later
Set earn_ratio to 10                     // Min %10 profit
Set down  to 1                           // Trend of stock
Set up to 2                              // Trend of stock
Set up_limit to 5                        // Threshold
Set up_limit_Increase to 2               // Threshold
Set Index_list to  []                    //Index of stock_DataFrame
Set counter_list to []                   // Go back days before crosses EMA
Set Index_Counter to []                  // concatenate list
Set Index_Finish to 0                    // Define index
Set counter to 0                         // Transaction days
Set counter_up to 0                      // Flexibility of system
Set start to 0                           // Start index of stock
Set finish to 0                          // Last index of stock

FOR Each stock_code in the list stock_list:
    stock_DataFrame = Call  load_stock_data(stock_code)
    finish ←  lenght(stock_DataFrame)
    FOR  i=0  to  Call range(start,finish):
        sign  ←  stock_DataFrame [i]["sign"])
        IF  sign = down THEN
                Increment counter
        ELSEIF  (sign = up) AND (counter > up_limit)AND (counter_up < up_limit_Increase)
                INCREMENT counter_up
        ELSE    (sign = up) AND (counter > up_limit)
                Index_Finish ←stock_DataFrame[i]["Index"]
                Index ADD Index_Finish
                counter_list ADD counter
                Counter ← 0
                Counter_up ←  0
        ENDIF
        Index_Counter ADD counter_list,Index
    END FOR
END FOR
FOREACH Index In Index_Counter
    start_last_value  ← stock_DataFrame[Index]
    max_last_value  ← stock_DataFrame[Index:Index INCREMENT with_in_day]
    profit ← (max_last_value - start_last_value)/start_last_value
    IF profit > earn_ratio
        Result  ← 'Earning Class'
    ELSE
        Result  ← 'Earning Trap Class'
    END IF
END FOREACH
```

Algorithm 2: Labeling Losing Class

```
Set with_in_day to 15                          // Max 15 days later
Set lost_ratio to 10                           // Min %10 lost
Set down  to 1                                 // Trend of stock
Set up to 2                                    // Trend of stock
Set up_limit to 5                              // Threshold
Set down_limit_Increase to 2                   // Threshold
Set Index_list to  []                          //Index of stock_DataFrame
Set counter_list to []                         // Go back days before crosses EMA
Set Index_Counter to []                        // concatenate list
Set Index_Finish to 0                          // Define index
Set counter to 0                               // Transaction days
Set counter_down to 0                          // Flexibility of system
Set start to 0                                 // Start index of stock
Set finish to 0                                // Last index of stock

FOR Each stock_code in the list stock_list:
    stock_DataFrame = Call  load_stock_data(stock_code)
    finish ←  lenght(stock_DataFrame)
    FOR  i=0  to  Call range(start,finish):
        sign   ←   stock_DataFrame [i]["sign"]
        IF  sign = up THEN
                Increment counter
        ELSE IF    (sign =down) AND(counter > up_limit)AND(counter_down <down_limit_Increase)
                INCREMENT counter_down
        ELSE       (sign = down) AND (counter > up_limit)
                Index_Finish ←stock_DataFrame[i]["Index"]
                Index ADD Index_Finish
                counter_list ADD counter
                Counter ← 0
                Counter_down ←  0
        ENDIF
        Index_Counter ADD counter_list,Index
    END FOR
END FOR
FOREACH Index In Index_Counter
    start_last_value  ← stock_DataFrame[Index]
    min_last_value   ← stock_DataFrame[Index:Index INCREMENT with_in_day]
    lost ← (start_last_value- min_last_value )/start_last_value
    IF lost > lost_ratio
        Result  ← 'Lossing Class'
    ELSE
        Result  ← 'Lossing Trap Class'
    END IF
END FOREACH
```

3.5 Determining Appropriate Technical Indicator Features

In this part of the study, the purpose is to determine the suitable financial technical indicators using the wrapper algorithms. To select the best possible performance with a specific learning algorithm on a training dataset, a feature subset selection algorithm should consider how the algorithm and the training set react. In our study, we use recursive feature elimination and sequential feature selection algorithms. These models choose a subset of features and measure the performance (accuracy) of the adopted set of features by utilizing a classifier. Therefore, wrappers can generate better feature subsets to improve the performance of the classification methods. Both approaches concentrated on the utilization of classification algorithms which are the regression tree (CART), and the gradient boosting algorithm (GBM) to control classification penalty. The first feature selection method RFE uses CART estimator, whereas the second feature selection method SFS applies GBM to evaluate the fitness of the identified feature subset. In this study, we applied both RFE and SFS methods and merged the features identified by the two methods to get a better result.

3.5.1. Recursive Feature Elimination (RFE) by using Classification and Regression Tree (CART)

RFE is a feature selection algorithm that removes the weakest feature until the optimum number of features is obtained. Features are sorted by the importance of the model's feature, and by recursively considering smaller and smaller sets of features per loop, RFE aims to remove unnecessary features based on variance and the correlation between them. There are two significant configuration choices when we define the best feature set using RFE: the choice in the number of features to choose and the selection of the method used to support selected features. The RFE approach to defining better feature subsets is handled by the use of classification and regression tree (CART) estimators. There is no particular reason for us to choose the CART method. The main reason for selecting CART is to use it for both classification and regression problems. In our system, our model makes classification therefore CART can be used to select best features for the classification process. Fig 3.12 presents how RFE selects best feature subset.

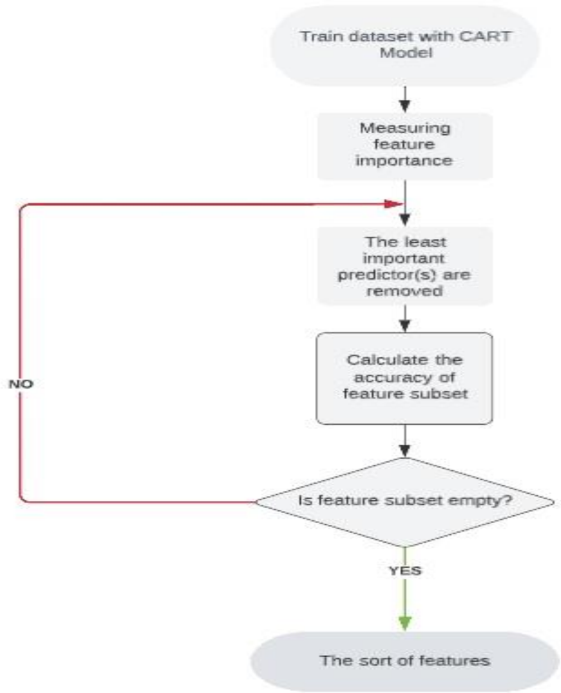


Figure 3.12. Presentation of RFE using CART.

As a result, after the loop is ended the feature subset size is a tuning parameter for RFE. The feature subset size that increases the performance criteria is used to adopt the features based on the importance rankings. The final subset is then used to train the system.

There are 19 technical indicators selected by RFE and they are listed in Table 3.3. The reason for selecting these technical indicators is they have stable values and get a certain range of values. For example, RSI gets values between 0 and 100. It does not change from financial instrument to financial instrument.

Table 3.3. Technical Indicators Selected By Rfe

No	Indicator (Input)	Description
1	DI+	Positive Directional Indicator
2	DI-	Negative Directional Indicator
3	MFI	Money Flow Index
4	MFI_EMA_9	9-days EMA of MFI
5	MFI_MA_3	3-days MA of MFI
6	RSI	Relative Strength Index
7	RSI_MA_3	3-days SMA of RSI
8	RSI_MA_14	14-days SMA of RSI
9	RSI_EMA_9	8-days EMA of RSI
10	WR	Larry William %R
11	PPO_histo	PPO- PPO_histo
12	PPO	Percentage Price Oscillator
13	PPO_signal	9-days WMA of PPO
14	CCI	Commodity Channel Index
15	CMO	Chande Momentum Oscillator
16	MACD	Moving Average Convergence Divergence
17	MACD_signal	9-days WMA of MACD
18	ROC_9	9-days EMA of ROC
19	CMFI	Chaikin Money Flow Index

Table 3.4. The Accuracy Values of Each Subset of Selected Technıcal Indicators By Rfe With Cart Classifier

No	Indicator (Input)	Accuracy (%)
1	CMO	30.19%
2	CMO, MACD	30.80%
3	MFI, CMO, MACD	31.98%
4	MFI, PPO_histo, CMO, MACD	35.25%
5	MFI, WR, PPO_histo, CMO, MACD,	33.46%
6	MFI, WR, PPO_histo, CMO, MACD, ROC_9	34.39%
7	MFI, WR, PPO_histo, CMO, MACD, ROC_9, CMFI	33.46%
8	DI+, MFI, WR, PPO_histo, CMO, MACD, ROC_9, CMFI	34.00%
9	DI+, DI-, MFI, WR, PPO_histo, CMO, MACD, ROC_9, CMFI	34.16%
10	DI+, DI-, MFI, WR, PPO_histo, PPO, CMO, MACD, ROC_9,CMFI	34.78%
11	DI+, DI-, MFI, RSI_MA_14, WR, PPO_histo, PPO, CMO,MACD, ROC_9, CMFI	35.56%
12	DI+, DI-, MFI, RSI_MA_14, WR, PPO_histo, PPO, CCI,CMO, MACD, ROC_9, CMFI	35.01%
13	DI+, DI-, MFI, RSI, RSI_MA_14, WR, PPO_histo, PPO,CCI, CMO, MACD, ROC_9, CMFI	34.63%
14	DI+, DI-, MFI, MFI_EMA_9, RSI, RSI_MA_14, WR, PPO_histo,PPO, CCI, CMO, MACD, ROC_9, CMFI	36.65%
15	DI+, DI-, MFI, MFI_EMA_9, RSI, RSI_MA_14, WR, PPO_histo,PPO, PPO_signal, CCI, CMO, MACD, ROC_9, CMFI	35.25%
16	DI+, DI-, MFI, MFI_EMA_9, RSI, RSI_MA_14, RSI_EMA_9, WR,PPO_histo, PPO, PPO_signal, CCI, CMO, MACD, ROC_9,CMFI	35.40%
17	DI+, DI-, MFI, MFI_EMA_9, RSI, RSI_MA_14, RSI_EMA_9, WR,PPO_histo, PPO, PPO_signal, CCI, CMO, MACD, MACD_signal,ROC_9, CMFI	34.94%
18	DI+, DI-, MFI, MFI_EMA_9, MFI_MA_3, RSI, RSI_MA_14,RSI_EMA_9, WR, PPO_histo, PPO, PPO_signal, CCI, CMO,MACD, MACD_signal, ROC_9, CMFI	36.80%
19	*DI+, DI-, MFI, MFI_EMA_9, MFI_MA_3, RSI, RSI_MA_3,RSI_MA_14, RSI_EMA_9, WR, PPO_histo, PPO, PPO_signal, CCI,CMO, MACD, MACD_signal, ROC_9, CMFI*	*36.73%*

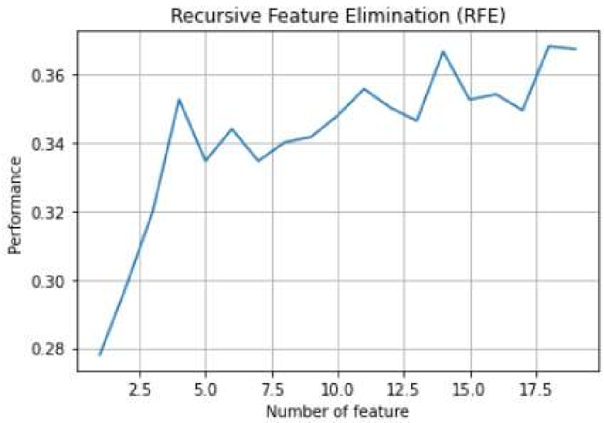


Figure 3.13. Effective feature representation of RFE

Table 3.4 presents accuracy of classification when the selected feature subsets are used. As shown in Table 3.4 and Figure 3.13 the highest accuracy at 36.80% is obtained when 18 features are used. These selected features are *DI+, DI-, MFI, MFI_EMA_9, MFI_MA_3, RSI, RSI_MA_14, RSI_EMA_9, WR, PPO_histo, PPO, PPO_signal, CCI, CMO, MACD, MACD_signal, ROC_9, CMFI*.

3.5.2 Sequential Feature Selection (SFS) by using Gradient Boosting Algorithm (GBM)

Sequential feature selection algorithms are a family of greedy search algorithms which are used to reduce an initial n-dimensional feature space to a k-dimensional feature subspace where k < n. The reasoning behind feature selection algorithms is to adopt a subset of features that is most appropriate to the problem, which results in optimal computation efficiency and diminishes the generalization error of the model by eliminating irrelevant features. SFS is an iterative method evaluating the performance metric and calculating a statistical significance test in a for-loop, training each iteration model with a different number of features and then measuring how the system performs (Ferri et al., 1994) . The SFS system follows the below steps.

In this algorithm, at each iteration one feature x^+ is selected and it is inserted into the feature subset X_t, x^+ is the feature that optimizes the selection function, that is, the feature that is involved with the best feature performance.

When the system determines the best feature using SFS: the choice in the number of features to choose and the selection of the method used to help adopt

Input: $X = \{x_1, x_2, \dots, x_n\}$ The SFS algorithm takes the n features as input

Output: $X_t = \{x_i \mid i = 1,2, \dots, t;\ x_i \in X\}$ where $t = 1,2, \dots, k$; the return of subset features; where k< n

Initialization: $X_0 = \emptyset, \mathrm{t} = 0$

Iter 1:

$X^+ = arg$ max I $(X_t + X)$, where $x \in X - X_t$

$X_{t+1} = X_t + x^+$

t=t+1

Go to Iter 1

Terminate k=p

features to be done. The SFS approach to defining better feature subsets is handled by the use of the Gradient Boosting Machine (GBM) predictor. The reason for choosing GBM is to optimize different loss functions and offers many hyperparameter tuning alternatives that make the function regulation very flexible. Fig 3.13 shows flowchart of SFS approach using GBM.

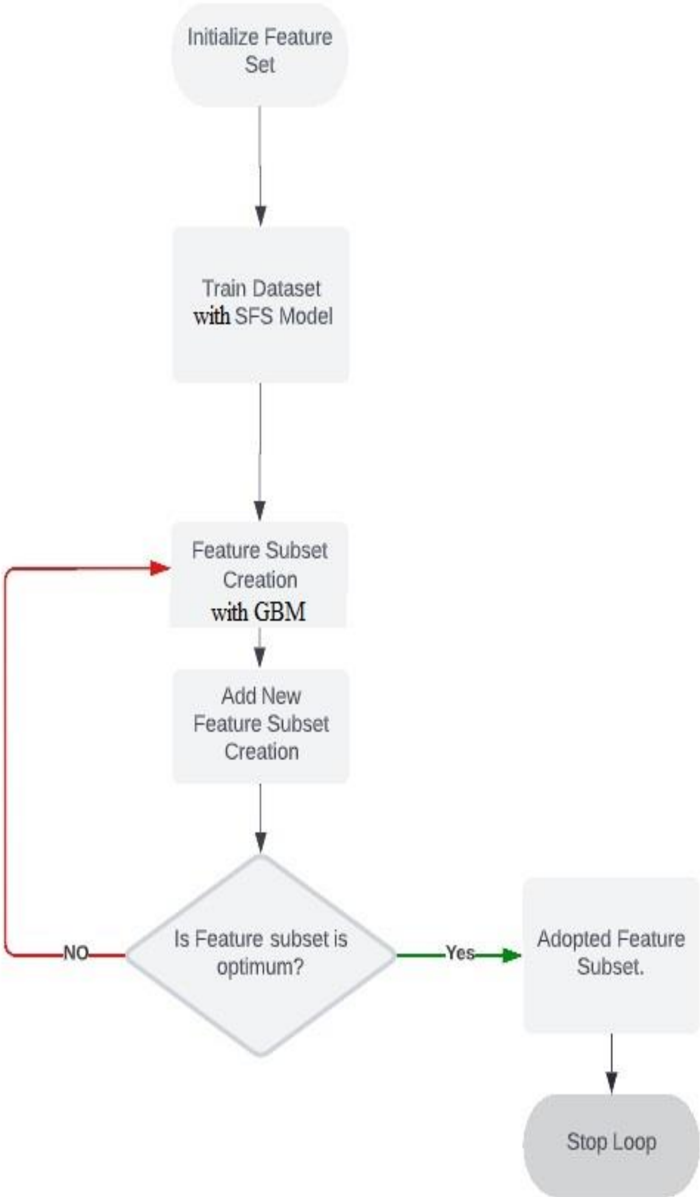


Figure 3.14. Presentation of SFS using GBM.

Table 3.5. The Accuracy Result of Technical Indicators Used in SFS Wıth GBM

No	Indicator (Input)	Accuracy(%)
1	CMO	37.66%
2	MFI, CMO	38.17%
3	DI+, MFI, CMO	38.69%
4	DI+, MFI, CMO, MACD_signal	38.54%
5	DI+, MFI, RSI_MA_14, CMO, MACD_signal	38.75%
6	DI+, MFI, RSI_MA_14, PPO, CMO, MACD_signal	38.54%
7	DI+,MFI,RSI_MA_14,PPO,CCI,CMO,MACD_signal	39.12%
8	DI+,MFI, RSI_MA_14,PPO_histo, PPO,CCI,CMO,MACD_signal	39.18%
9	DI+,MFI,RSI_MA_14,PPO_histo,PPO,CCI,CMO,MACD, MACD_signal	39.55%
10	DI+,MFI,RSI_MA_14,WR,PPO_histo,PPO,CCI,CMO,MACD,MACD_signal	39.60%
11	DI+,DI,MFI,RSI_MA_14,WR,PPO_histo,PPO,CCI,CMO, MACD,MACD_signal	39.66%
12	DI+,DI,MFI,RSI,RSI_MA_14,WR,PPO_histo,PPO,CCI,CMO,MACD,MACD_signal	39.12%
13	DI+,DI,MFI,RSI,RSI_MA_14,WR,PPO_histo,PPO,PPO_signal,CCI,CMO,MACD,MACD_signal	39.22%
14	DI+,DI,MFI,RSI,RSI_MA_14,RSI_EMA_9,WR,PPO_histo,PPO,PPO_signal,CCI,CMO,MACD,MACD_signal	39.29%
15	DI+,DI,MFI,MFI_MA_3,RSI,RSI_MA_14,RSI_EMA_9, WR,PPO_histo,PPO,PPO_signal,CCI,CMO,MACD,MACD_signal	39.47%
16	DI+,DI,MFI,MFI_MA_3,RSI,RSI_MA_14,RSI_EMA_9, WR,PPO_histo,PPO,PPO_signal,CCI,CMO,MACD,MACD_signal,CMFI	39.22%
17	DI+,DI,MFI,MFI_MA_3,RSI,RSI_MA_3,RSI_MA_14,RSI_EMA_9,WR,PPO_histo,PPO,PPO_signal,CCI,CMO,MACD,MACD_signal,CMFI	38.98%
18	DI+,DI,MFI,MFI_MA_3,RSI,RSI_MA_3,RSI_MA_14,RSI_EMA_9,WR,PPO_histo,PPO,PPO_signal,CCI,CMO,MACD,MACD_signal,ROC_9,CMFI	39.06%
19	DI+,DI,MFI,MFI_EMA_9,MFI_MA_3,RSI,RSI_MA_3,RSI_MA_14,RSI_EMA_9,WR,PPO_histo,PPO,PPO_signal,CCI,CMO,MACD,MACD_signal,ROC_9,CMFI	37.66%

Table 3.5 and Figure 3.15 presents classification accuracy values of the GBM classifier when feature subsets selected by SFS are used. As shown in Table 3.5, the highest accuracy score at 39.66% is obtained with 11 features using the GBM classifier. The selected features are *DI+, DI-, MFI, RSI_MA_14, WR, PPO_histo, PPO, CCI, CMO, MACD, and MACD_signal*.

Both feature selection methods select different sets of features, so we decided to combine the set of features selected by both methods. Therefore, we used the following feature list that contains *DI+, DI-, MFI, MFI_EMA_9, MFI_MA_3, RSI, RSI_MA_14, RSI_EMA_9, WR, PPO_histo, PPO, PPO_signal, CCI, CMO, MACD, MACD_signal, ROC_9, CMFI* features in our study.

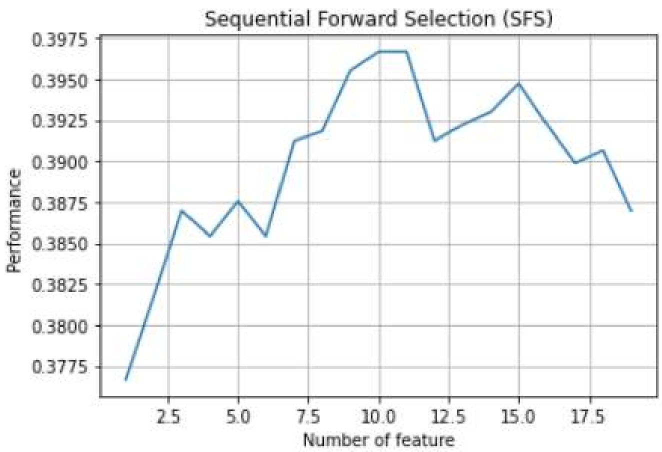


Figure 3.15. Effective feature representation of SFS

3.6. Generation of Input Dataset

In the previous section 3.5, with RFE using CART model and SFS using GBM model optimal features are obtained. The RFE achieved the best accuracy score with the CART model using 18 features. The SFS reached the best accuracy score with 11 features using the GBM method. After defining the optimal features, these features are merged with our proposed DBEMA features, and we have 24 features in total. In Table 3.6 the features that are used are illustrated.

Table 3.6. The List of Feature Sets

No	Model	Count	Feature list
1	RFE using the CART model	18	DI+, DI-, MFI, MFI_EMA_9, MFI_MA_3, RSI, RSI_MA_14,RSI_EMA_9, WR, PPO_histo, PPO, PPO_signal, CCI, CMO, MACD, MACD_signal, ROC_9, CMFI
2	SFS using GBM model	11	DI+, DI-, MFI, RSI_MA_14, WR, PPO_histo, PPO, CCI, CMO, MACD, MACD_signal.
3	A Novel DBEMA	6	EMA9_EMA50_dist_sign, EMA9_EMA50_dist, EMA9_EMA200_dist_sign, EMA9_EMA200_dist, EMA50_EMA200_dist_sign, EMA50_EMA200_dist.
4	Union of Features	24	DI+, DI-, MFI, MFI_EMA_9, MFI_MA_3, RSI, RSI_MA_14,RSI_EMA_9, WR, PPO_histo, PPO, PPO_signal, CCI, CMO, MACD, MACD_signal, ROC_9, CMFI, EMA9_EMA50_dist_sign, EMA9_EMA50_dist, EMA9_EMA200_dist_sign, EMA9_EMA200_dist, EMA50_EMA200_dist_sign, EMA50_EMA200_dist.

The 24 features listed in the last row of Table 3.6 are used to generate a 2-D image-like array for each stock. The above-listed parameters were computed by using FinanceLab, which was developed in the Python programming language. These optimal features and the novel DBEMA features were designed for forecasting trends of financial time series over short, medium, and long time periods. The period of the indicators was 9, 14, 50, and 200 days in this study. Each image had a 75×24 dimensions generated by using 24 features for 75 days period. At the same time, each image was labeled as earning class, earning trap class, losing class, or losing trap class by using the Algorithm 1 and Algorithm 2. The sorting of the features, before image creation, was necessary; related indicators should come together to produce meaningful image records, so we gathered the technical indicators common to the SFS and RFE feature lists and DBEMA parameters by their typical properties, such as their manner indicators.

In this study, 6425 images from BIST data (between June 1, 1998, and April 21, 2021) were created for training and testing a CNN. The BIST data contains the 2001 and 2008 financial crisis periods. Class distributions of images are as follows: earning class has 1953 samples, earning trap class has 1244 images, losing class has 2011, and losing trap class has 1217 samples. Moreover, out of 6425 images, 5140 of them were used as training the CNN and the remaining 1285 images were used for testing. The order of features used are given in Figure 3.16.

The other important point of selecting image size is the number of days back. In this study, several periods including 50, 75, 100, and 125 days were experimentally evaluated. The results of these experiments are given in Table 3.7. As it can be seen in Table 3.7, the system gets the best accuracy when it works for 75 days period. Therefore 75 days values of the 24 features are used to form images that's why image size is 75 * 24. We also have 1 additional feature which is the class label of the image.

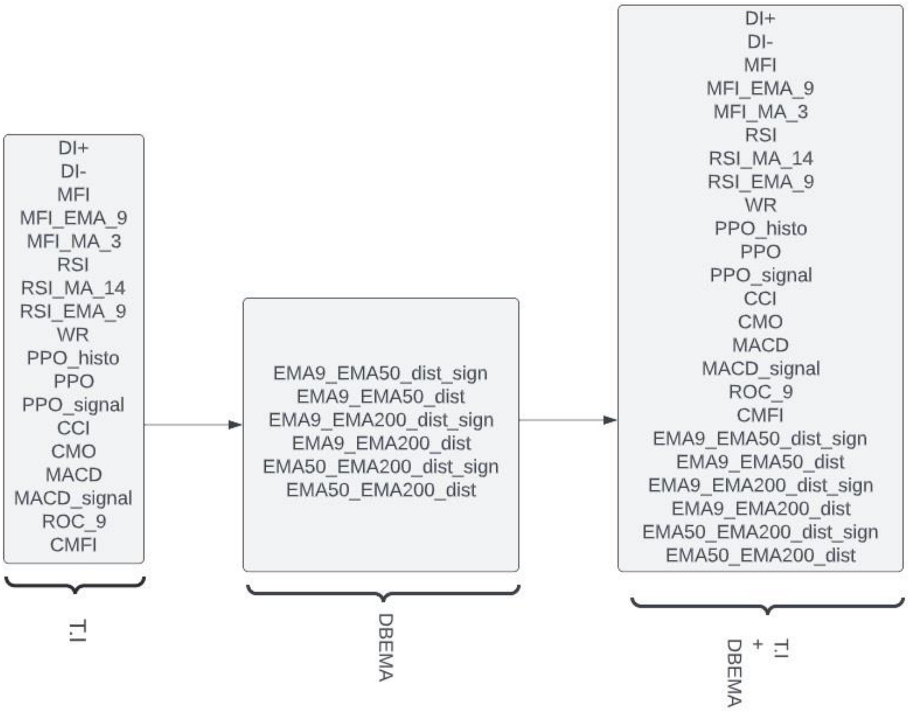


Figure 3.16. 2-D-image-like representation of a stock.

Table 3.7. The Accuracy Result of Number of Days Back USED CNN-DBEMA

No	Image Count	Method	Number of days back	Accuracy (%)
1	6425	CNN-DBEMA	50-days-period	56.11%
2	6425	CNN-DBEMA	75-days-period	57.28%
3	6425	CNN-DBEMA	100-days-period	52.69%
4	6425	CNN-DBEMA	125-days-period	53.58%

For the robustness of comparability, the three different representations of the image dataset were used for financial time series prediction using the benchmarking machine learning models. Fig. 3.17, Fig. 3.18, Fig. 3.19 show different representations of the dataset used. The reason for generating different representations of the dataset is using different techniques for financial time series analysis and each technique may perform better in different representation. For example, CNN architectures work with two-dimensional array type data for each data instance. However, other machine learning algorithms such as random forest and kNN require a vector for each instance in the dataset.

In Fig. 3.17, the first representation method, which contains 2-d array of size 75*24 for each data instance, is displayed. This dataset form is used to experimentation with CNN architectures. In Fig. 3.18, the second representation method is shown in which only the last row (the feature values for the 75th day which belong to the reference point (Cross-over) is used to represent a data instance. In Fig 3.19 the third representation method is shown in which average of 75-days feature values are used. In that case each data instance is shown by a vector of size 24. In Fig 3.20, all 75 days feature values are used but the 75*24 matrix is converted into a vector by concatenating each row so that the vector has 75*24 = 1800 elements. The last three representations are used to test benchmark classifiers other than CNN based methods.

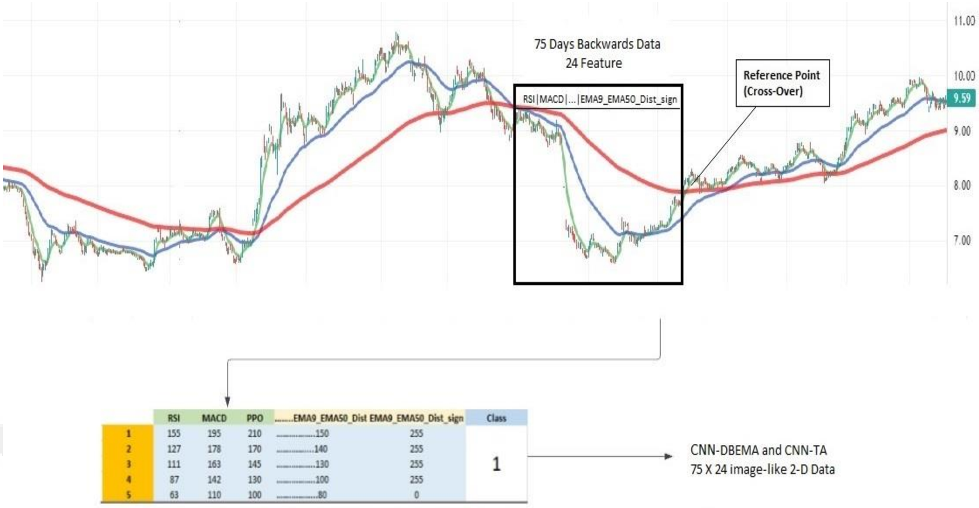


Figure 3.17. Representation CNN's Input Data.

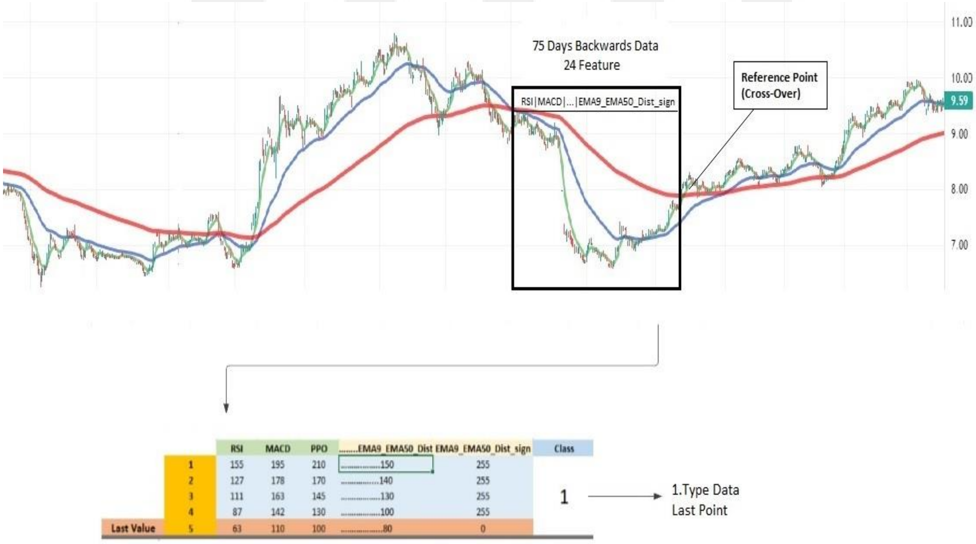


Figure 3.18. Representation of the Last Day of the 75-Days Input Data

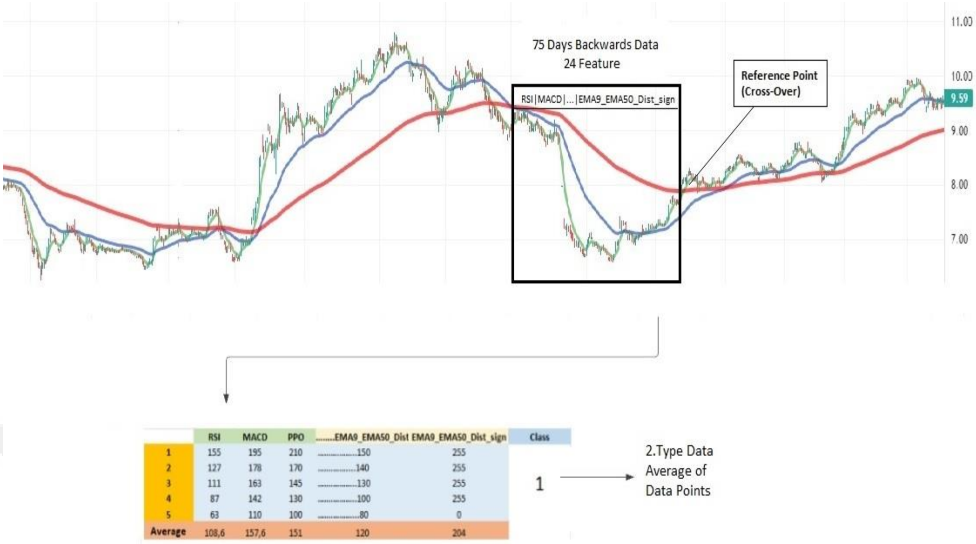


Figure 3.19. Representation of the Average of the 75-Days Input Data

Figure 3.20. Representation of All Data of the 75-Days Input Data

3.7. The Benchmarking Models

In this thesis we used the well-known machine learning techniques such as linear regression (LR), bagged trees regressor (BAG), Gaussian Naive Bayes (N.B.), k-nearest neighbors (KNN), random forests (R.F.), multi-layer perceptron (MLP), Convolutional Neural Network- Technical Analysis (CNN-TA), relative strength index method (RSI) that are widely used in stock market predictions to make comparison with our proposed method. The parameters used by each ML model are shown in Table 3.10.

3.7.1 Linear Regression (LR) Method

When we implement a linear regression model for our data set, independent features are $x = (x_1, x_2, \ldots.x_t)$, and y is the dependent feature, $y = S_{0+}S_1x_1 + S_2x_2 + \cdots S_tx_t + ¢$ denotes the relation between y and x where t is the number of features, $S = (S_1, S_2, \ldots.S_t)$ $S_1, S_2, \ldots.S_t$ are the regression coefficients and $¢$ is the error. In this equation, n is the number of training samples.

The mathematical representation of the multivariate regression problem is as follows;

[Feature matrix] * [Coeficients Vector]=[Class vector]

$$\begin{bmatrix} X_{11} & \cdots & X_{1t} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nt} \end{bmatrix} X \begin{bmatrix} S_1 \\ S_2 \\ . \\ . \\ . \\ S_t \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix} \tag{3.40}$$

3.7.2 Bagged Trees Regressor (BAG) Method

The bagged Trees Regressor (BAG) method is a general method for fitting multiple editions of a prediction model and then ensembling them into an aggregated prediction. Bagging can be used for regression or classification.

Mathematically, bagging is shown by the below expression in Eq. 3.41. The left side of the equation is bag prediction and on the right side of the equation are the individual predictors.

$$f_{bag} = f_1(x) + f_2(x) + \cdots + f_n(x) \tag{3.41}$$

3.7.3 K-Nearest Neighbors (KNN) Method

The KNN algorithm is a supervised machine learning model used for both classification and regression problems. In this study, the model is implemented by best parameters to achieve the highest accuracy score. In Table 3.8 the best parameters used in this study are illustrated.

Table 3.8. The Knn Tuning Parameters Used in Predıctıon

No	Parameter Name	Value
1	leaf_size	30
2	metric	Minkowski
3	n_neighbors	48
4	*GridSearchCV*	*10*

3.7.4 CNN-TA

CNN-TA is a CNN based model for financial time series analysis (Sezer and Ozbayoglu 2018). The layers of the CNN-TA model are as follows: an input layer, two convolutional layers, a max-pooling layer, two dropout layers, and a fully connected layer and an output layer. Each image is a 2D array with a size equal to 15 x 15. Each image was generated by using 15 days period and 15 technical indicators. The convolutional layers, respectively, had 32 and 64 neurons and outputs of 15 x 15 x 32 and 15 x 15 x 64 dimensions. The max-pooling layer had dimensions of 7 x 7 x 64. The dropout values were 0.25 and 0.50. The last layer of the model, which is the fully connected layer, had 128 neurons. Four outputs were generated by the CNN-TA model. This classes are “earning class”, “earning trap class”, “losing class”, and “losing trap class”. In this model, a 3 x 3 size CNN filter/kernel was used. To achieve outputs, a softmax function was used. The model was implemented with the Python programming language using the Keras and TensorFlow libraries.

3.7.5 Multi-Layer Perceptron (MLP) Method

Multi-layer Perceptron (MLP) is a supervised machine learning method that has a wide usage for classification and regression applications in many financial areas (Kamalov et al., 2021). MLP focuses on the function $f(x): R^n \rightarrow R^o$ which learns by training on a dataset, where n is the number of features (inputs) and o is the number of classes (outputs). $x = (x_1, x_2, \dots . x_n)$ are features and y is the target, $w = (w_1, w_2, \dots . w_n)$ are weights.

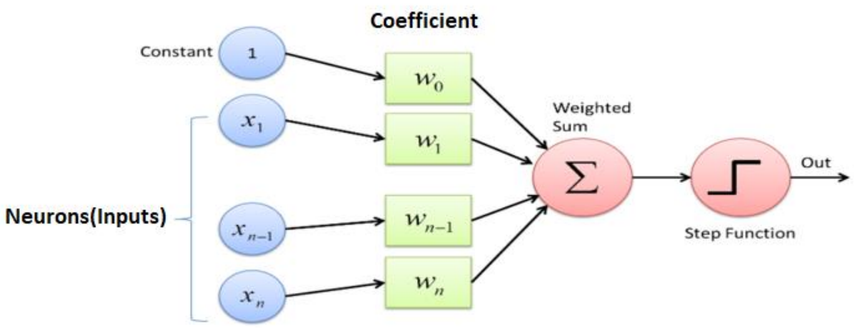


Figure 3.21. Presentation of a Perceptron in MLP

In this study, the parameters of MLP used are as follows: hidden_layer_sizes is 100, activation function is relu, solver is adam, batch_size is 200, learning_rate_init is 0.001, max_iter is set to 200. hidden_layer_sizes represent the number of neurons in the ith hidden layer; activation function determines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network; solver is used for weight optimization; alpha is L2 penalty (regularization term) parameter; batch_size gives the size of batches for stochastic optimizers; learning_rate_init controls the step-size in updating the weights; and max_iter is the maximum number of iterations.

In Fig 3.21, a neuron of an MLP is represented. In the figure, w_0 represent the bias unit. In the first step, inputs are fed into the neuron and multiplied by their weights. Then the w_0 value and the weighted input values are added for feeding the sum through the activation function. The result of activation function is the output of the neuron. The same operations happen to other neurons get inputs and gather an initial set of random weights. These are merged in weighted sum and the activation function defines the output.

3.7.6 Random Forests (RF) Method

Random forest (RF) is one of the most popular and extensively used machine learning algorithms given its magnificent performance across a large range of classification and regression problems. Random forest is an ensemble machine learning algorithm that fits many decision tree predictors on various sub-samples of the dataset and gets averaging to enhance the predictive accuracy. In Fig. 3.22 the random forest method is presented. In this study, the RF model has n_estimators=100.

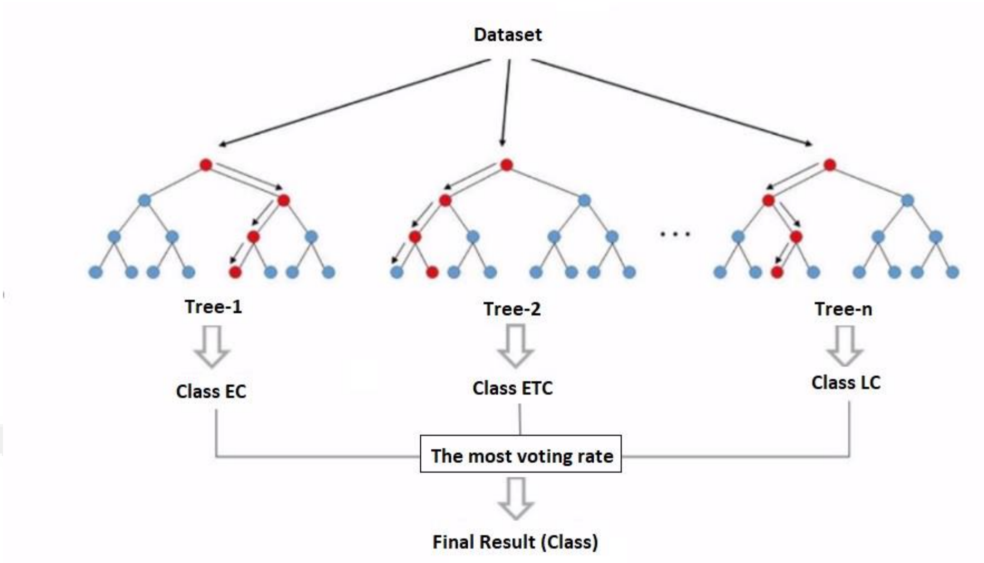


Figure 3.22. Presentation of Random Forrest

3.7.7 Relative Strength Index Method

The relative strength index (RSI) is a financial technical indicator that evaluates overbought-oversold signals. It determines the imbalances in the up and down movements of a price and the normalization rate of these imbalances. Using this determination process produces buy-sell signals for the researcher. The most common strategy in the market is to buy when a 14-day period RSI falls below 30 and sell when it increases above 70 (Chen et al., 2021). Eq. 3.41 displays the formula for the RSI. The *upwardReturn* parameter of the stock shows if the close price is higher than the previous price. The *downwardLost* parameter of the stock shows if the close price is lower than the previous price.

$$RSI = 100 - \frac{100}{1 + (upwardReturn/downwardLost)} \tag{3.41}$$

3.8 Proposed CNN-DBEMA Model

A CNN involves multi-dimensional layers and a set of kernels onto which the convolution process is applied. The connection between the layers is the basic difference between a CNN and a NN. In CNN architecture, a receptive area of the input arrays is connected to only one neuron; in contrast, the flatter input arrays in an NN. are fully connected to the neurons. Convolutional neural networks apply a kernel on one-dimensional or multi-dimensional input to create a feature map that discovers existing features in the input. In a convolution layer, the kernel moves on every possible position in the input. Element-wise multiplication between the arrays then is summed. After a convolution layer, outputs coming from this layer pass through an activation function. Then max-pooling decreases every four neurons to a unique one and adopts the highest value in subsampling from the activated outputs field (Ngiam et al., 2010).

In this thesis, the trend of the closing price movement of stocks was forecasted with our proposed CNN-DBEMA model. An illustration of the CNN-DBEMA is given in Fig. 3.22. Our initial input image size is 75 x 24. Two convolutional layers with sizes 32 x 75 x 24 and 32 x 35 x 10 are used. Two max-pooling layers with sizes 32 x 71 x 20 and 64 x 33 x 8 are used separately and three dropouts (0.3, 0.3, 0.3) are used to prevent overfitting. After that, layers are then connected to two fully connected layers have 256, and 128 neurons.

In the first convolution layer, we use 32 kernels on a 75 X 24 image-like 2D array with a size of 5 x 5 and a stride of 1. The activation function used at this layer is ReLU. The output feature map size is 32 x 71 x 20. Next, a max-pooling layer with a kernel size of 2 x 2 and a stride of 2 are used. The output feature map size is 32 x 35 x 10. The pooling layer does not change the number of channels. The second convolution layer has 64, 3 X 3 filters and a stride of 1. The activation function is also ReLU. The output size is 64 x 33 x 8. Again, the max-pooling layer size is 2 x 2 with a stride of 2. Thus, the size of the output is reduced to 64 x 16 x 4. After this, the CNN layers are fully connected with 128 neurons, which results in the output of

4 class values with the activation function ReLU. At the end of the system, the output layer passes through a softmax function with 4 neurons. The softmax function turns back the probability that an output belongs to a particular class. The details of our model are presented in Table 3.9.

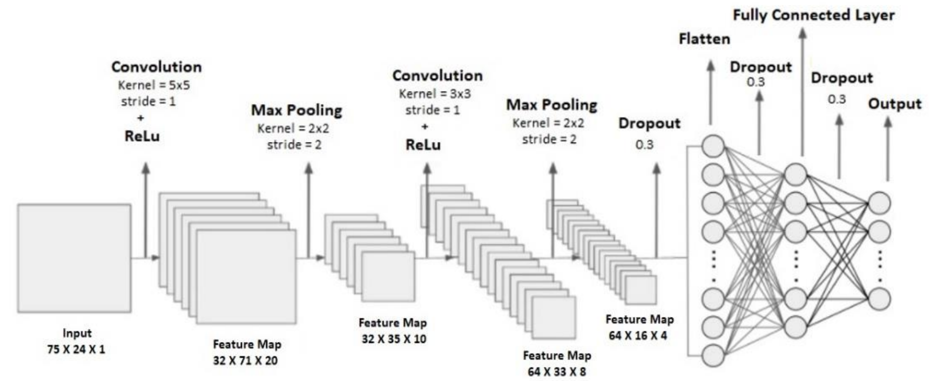


Figure 3.23Figure 3.23. CNN-DBEMA illustration.

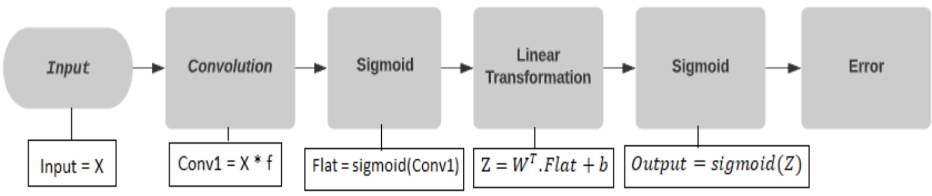


Figure 3.24. CNN-DBEMA flow diagram.

Table 3.9. Details of The Proposed Model

Layer	***Filter***	***Size***	***Stride***	***Output***	***ActivationFunction***
Input	-	-	-	1 x 75 x 24	
Conv 1	32	5 x 5	1	32 x 71 x 20	ReLu
Max Pooling 1		2 x 2	2	32 x 35 x 10	
Conv 2	64	3x 3	1	64 x 33 x 8	ReLu
Max Pooling 2		2 x 2	2	64 x 16 x 4	
Flatten				1024	
Fully connected layer 1	256			256	ReLu
Fully connected layer 2	128			128	ReLu
Output				*4*	*Softmax*

We applied a convolution operation to the CNN-DBEMA model as shown in Fig. 3.24 and Eq. 3.42, where X is n*m input matrix $X = \{x_{11,}x_{21,}\ldots, x_{nm,}\}$, f is defined as a filter/kernel, and w is defined as a randomly initialized weight matrix $W = \{w_{11,}w_{21,}\ldots, w_{n1,}, \ldots, w_{nm,}\}$, m must be equal to the number of features or inputs. B is defined as bias where $B = \{b_{1,}b_{2,}\ldots, b_{n,}\}$. In addition to taking the convolutional and max-pooling layers through a sequence of the deep neural network structure, Eq. 3.46 supplies details about the neural network. A softmax function provides the output in Eq. 3.47.

Table 3.10. The Features Used By Machine Learning Models

No	Parameters	LM	NB	KNN	RF	MLP	CNN-TA	CNN-DBEMA
1	SMA						+	
2	EMA						+	
3	RSI	+	+	+	+	+	+	+
4	MFI	+	+	+	+	+		+
5	ROC						+	
6	MFI_EMA_9	+	+	+	+	+		+
7	RSI_EMA_9	+	+	+	+	+		+
8	TR							
9	ATR							
10	DI+	+	+	+	+	+	+	+
11	DI-	+	+		+	+	+	+
12	DMI						+	
13	CMFI	+	+	+	+	+		+
14	TP							
15	TRIX	+	+	+	+	+		+
16	MACD	+	+	+	+	+		
17	VAMA	+	+	+	+	+		+
18	PSAR						+	
19	TEMA						+	
20	PPO	+	+	+	+	+	+	+
21	CMO	+	+		+	+	+	+
22	WMA						+	
23	HMA						+	
24	CCI	+	+	+	+	+	+	+
25	WR	+	+	+	+	+	+	+
26	RSI_MA_14	+	+	+	+	+		+
27	MFI_MA_3	+						
28	PPO_histo	+	+	+	+	+		+
29	PPO_signal	+	+	+	+	+		+
30	MACD_signal	+	+	+	+	+		+
31	ROC_9	+	+	+	+	+		+
32	EMA9_EMA50_dist	+	+	+	+	+		+
33	EMA9_EMA200_dist	+	+	+	+	+		+
34	EMA50_EMA200_dist	+	+	+	+	+		+
35	EMA9_EMA50_dist_sign	+	+	+	+	+		+
36	EMA9_EMA200_dist_sign	+	+	+	+	+		+
37	*EMA50_EMA200_dist_sign*	+	+	+	+	+		+

$$S(i,j) = (X * K)(i,j) \sum_m \sum_n X(m,n)K(i-m, j-n) \tag{3.42}$$

$$Conv1 = X * f \tag{3.43}$$

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.44}$$

$$Flat = sigmoid(Conv1) \tag{3.45}$$

$$\mathrm{Z} = W^T.Flat + b \tag{3.46}$$

$$y = softmax(Z) \tag{3.47}$$

4. RESULTS AND DISCUSSIONS

In this section, experimental results performed on CNN-DBEMA are given in detail and the experimental results are discussed. We also discuss some issues related to our proposed method CNN-DMEMA by comparing the experimental results based on computational performance evaluation and financial evaluation. The experiments were performed using an Intel Core i5-3470 CPU at 3.20 GHz and 16.0 GB RAM, with a NVIDIA GeForce GT 1030 graphics card to accelerate our entire PC experience. The methods were implemented in Python programming language using Tensorflow, OpenCV, Keras, Numpy, Theano, Sklearn libraries, as well as our developed library, which we called FinanceLab. We compared the performance of our proposed model with different categories of methods. These categories were machine learning methods, financial methods, and one of the new CNN based model called CNN-TA method.

4.1. Performance metrics for Evaluation of Classifiers

A confusion matrix is a measurement method for describing the performance of a classification model. In confusion matrix, rows demonstrate the actual classes. On the other hand, columns display the predicted classes. Our model's confusion matrix was applied to multi-class classification problem. As our problem is 4-classes classification problem, the size of the confusion matrix is 4 x 4, and the numbers of true classification cases are located on the diagonal of the matrix. The other values in the confusion matrix show either false negatives or false positives. The confusion matrix gives us prediction results and its inner operations for observing details among the classes.

Confusion matrix allows us to compute sensitivity, specificity, precision, accuracy, and f1-score metrics that are also used to compare performance of the models.

To reach a satisfactory result, models are compared these. Evaluation metrics that symbolize the achievement of a proposed model and other benchmarking models. An important part of the evaluation principle is their ability to distinguish among model results. The terms true positive (TP), true negative (TN), false positive (FP), and false negative (FN) from the confusion matrix are used for the calculation of performance metrics. True Positive Rate (TPR) is also called sensitivity, and False Positive Rate (FPR) is also called specificity. The evaluation of accuracy, precision, recall, and F-measure (F1) is performed using TP, TN, FP, and FN

In this study, the proposed model makes multiple class classification, hence performance evaluation formulas consider all four classes, and these formulas are described in Eqs. (4.1)–(4.9):

$$\sum_{i=0}^{nClasses} TP_i = TP_{1(\mathrm{E.C.})} + TP_{2(\mathrm{ETC})} + TP_{3(\mathrm{L.C.})} + TP_{4(\mathrm{LTC})} \tag{4.1}$$

$$\sum_{i=0}^{nClasses} FP_i = FP_{1(\mathrm{E.C.})} + FP_{2(\mathrm{ETC})} + FP_{3(\mathrm{L.C.})} + FP_{4(\mathrm{LTC})} \tag{4.2}$$

$$\sum_{i=0}^{nClasses} FN_i = FN_{1(\mathrm{E.C.})} + FN_{2(\mathrm{ETC})} + FN_{3(\mathrm{L.C.})} + FN_{4(\mathrm{LTC})} \tag{4.3}$$

$$\sum_{i=0}^{nClasses} TN_i = TN_{1(\mathrm{E.C.})} + TN_{2(\mathrm{ETC})} + TN_{3(\mathrm{L.C.})} + TN_{4(\mathrm{LTC})} \tag{4.4}$$

$$Sensivity = \frac{\sum_{i=0}^{nClasses} TP_i}{\sum_{i=0}^{nClasses} TP_i + \sum_{i=0}^{nClasses} FN_i} \tag{4.5}$$

$$Specificity = \frac{\sum_{i=0}^{nClasses} TN_i}{\sum_{i=0}^{nClasses} TN_i + \sum_{i=0}^{nClasses} FP_i} \tag{4.6}$$

$$Precision = \frac{\sum_{i=0}^{nClasses} TP_i}{\sum_{i=0}^{nClasses} TP_i + \sum_{i=0}^{nClasses} FP_i} \tag{4.7}$$

$$F1-score \quad = 2 * \frac{(Sensivity*Precision)}{(Sensivity+Precision)} \tag{4.8}$$

$$Acc = \frac{\sum_{i=0}^{nClasses} TP_i + \sum_{i=0}^{nClasses} TN_i}{\sum_{i=0}^{nClasses} TP_i + \sum_{i=0}^{nClasses} FP_i + \sum_{i=0}^{nClasses} FN_i + \sum_{i=0}^{nClasses} TN_i} \tag{4.9}$$

In the above equations, TP and TN are the numbers of correctly predicted positive and negative samples, whereas FP and FN indicate the number of incorrectly predicted positive and negative samples. The Acc is the ratio of the correctly predicted values. F-score, also known as F-measure, is a harmonic mean of precision and recall (a.k.a. sensitivity).

A ROC curve (receiver operating characteristic curve) is a figure demonstration of the trade-off between a classifier's sensitivity and specificity that is used for performance measurement. The ROC curve is generated by plotting the sensitivity (TP rate) on the y-axis against 1–specificity (FP rate) on the x-axis for the various tabulated values. The effective area is the top-left corner of the plot: FPs are 0 and TPs are 1. The higher the region under the curve (AUC), the better the model usually is. This curve denotes a trade-off between Sensitivity and Specificity, and it is valuable for realizing the overall accomplishment of the models.

4.2. Experimental Results of the Models and Their Comparison

In this section, the evaluation of the experimental results is presented. The proposed CNN-DBEMA model was compared with methods selected from common benchmarking machine-learning methods, deep learning methods, and financial methods. According to the predictions of the methods, each stock was revealed to 4 different situations. In the first case, the sample data was labeled 1, and the stock was bought at the short exponential moving average (EMA9), cutting the long exponential moving average (EMA200) upwards with the desired amount of the currently available capital over a predefined period (earning class). In the second case, the sample data was labeled 2, and the currently available capital was held, even if the short exponential moving average cut the long average upwards over a predefined period. Such a case is named an earning trap condition. In the third case, sample data was labeled 3, and the stock was sold at the short exponential moving average, cutting the long exponential moving average downwards. If the stock is held at this point, more money will be lost (losing class). In the last case, sample data was labeled 4, and the stock was held even if the short exponential moving average cut the long average downwards. Such a case is named a losing trap condition.

A novel proposed model was devised under specific conditions so that the model would be modifiable with tunable variables. Our model assumed a certain return acceptance and a loss acceptance in a predefined time frame. For the experimental analysis, we set system variables as at least a 10% return in 60 working days for the earn classes pattern and at least a 10% loss in 30 working days for the loss classes pattern. Our novel system is limited to these flexible variables.

The benchmark models and proposed model comparison results are exhibited in Table 4.1, Table 4.2, and Table 4.3. The financial time series trend forecasting tests were done by using the three representation of the dataset for the benchmarking machine learning models, and the ROC curves of the models are as shown in Fig. 4.1. Depending on Fig 3.4, we used data obtained from 75 days periods as input for CNN-DBEMA and CNN-TA. For other methods, three different data representations were applied: i) only the last day (crossover point) of the 75 days period ; ii) the average of each feature for the 75 days period; and iii) all data for the 75 days period in the 2D array are converted to a vector by concatenation of each row of the array.

Table 4.1 presents the performance of prediction accuracy of our proposed model versus the benchmarking models, which included crossover points (last day) of data. In terms of prediction accuracy, our proposed CNN-DBEMA model accomplished the second-highest score among the benchmarking models. Even though the CNN-DBEMA obtained the second-highest accuracy score, the novel proposed model (CNN-DBEMA) reached better sensitivity and specificity scores. However, the scores were very close to each other. For the purpose of checking the accuracy of the evaluation of the proposed CNN-DBEMA model, the second type of dataset was used that included the average values of the last 75 days of stocks.

Table 4.2 shows the performance of the proposed model versus the benchmarking models for the average values of the last 75 days of stocks. A third approach to validating the accuracy of our proposed model versus the benchmarking models containing all of the data points of the input matrices. Fig.4.1 denotes other performance determination metrics, involving sensitivity, specificity, F-score, and precision, that can evaluate how well models can label each data instance between different classes. In Table 4.2, it is obvious that the CNN-DBEMA model can reach the best forecasting performance. It is obvious that the predictive ability of the RSI model alone is very weak. RSI is a technical indicator, and it is only this indicator that does not make any sense on its own.

Class-based ROC examinations are presented in Fig. 4.1 and Fig. 4.2. In this comparison, the CNN-DBEMA predicted the earning class better than other methods. The prediction of both earn and loss trap classes are more difficult to predict the earn and loss classes. Comparing the overall performance of the methods, we observed that, the random forest and multi-layer perceptron methods had very similar results, as seen in Table 4.3. But the CNN-DBEMA model had better results.

Table 4.1. Performance Evaluation of Proposed Model (CNN-DBEMA) versus. Benchmarking Methods with Cross-Over (Last Day) Points of Data with DBEMA features.

No	Method Name	Acc_Scr (%)	Se (%)	Sp (%)	F1-score (%)	Precision(%)
1	Relative Strenght Index(RSI)	31.78%	--	-	--	--
2	Lineer Regression (LR)	36.58%	37.14%	77.42%	40.49%	45.61%
3	Bagged Trees Regressor(BAG)	41.87%	42.12%	80.16%	46.53%	51.98%
4	CNN-TA	43.95%	36.95%	80.69%	39.05%	41.71%
5	Gaussian Naive Bayes (NB)	51.98%	49.65%	83.74%	49.65%	49.73%
6	K-Nearest Neighbors (KNN)	52.61%	49.58%	83.82%	49.97%	50.36%
7	Random Forests (RF)	53.80%	52.60%	84.49%	53.86%	55.20%
8	CNN-DBEMA	57.28%	**57.43%**	**87.04%**	56.53%	55.67%
9	*Multi-Layer Perceptron (MLP)*	**57.42%**	57.29%	86.17%	**57.63%**	**57.97%**
10	Average (%)	47.47%	47.84%	82.94%	49.18%	51.02%

Table 4.2. Performance Evaluation of Proposed Model (CNN-DBEMA) versus Benchmarking Methods with Average of 75 Days Points of Data with DBEMA features.

No	Method Name	Acc_Scr (%)	Se (%)	Sp (%)	F1-score(%)	Precision(%)
1	Relative Strenght Index (RSI)	31.78%	--	--	--	--
2	Lineer Regression (LR)	31.64%	31.60%	77.38%	33.79%	35.57%
3	Bagged Trees Regressor(BAG)	34.40%	34.46%	76.24%	36.83%	39.57%
4	K-Nearest Neighbors (KNN)	41.17%	39.15%	76.83%	40.09%	40.15%
5	Gaussian Naive Bayes (NB)	41.17%	39.88%	79.21%	39.45%	39.53%
6	CNN-TA	43.95%	36.9%	80.6%	39.05%	41.71%
7	Multi-Layer Perceptron (MLP)	44.20%	41.74%	79.95%	41.76%	42.99%
8	Random Forests (RF)	45.14%	42.01%	78.42%	43.55%	45.22%
9	*CNN-DBEMA*	**57.28%**	**57.43 %**	**87.04 %**	**56.53%**	**55.67 %**
10	*Average (%)*	*41.19%*	*40.02%*	*79.45%*	*41.19%*	*42.55%*

Table 4.3. Performance Evaluation of Proposed Model (CNN-DBEMA) versus Benchmarking Methods with All of 75 Days Points of Data with DBEMA features.

No	**Method Name**	Acc_Scr (%)	Se (%)	Sp (%)	F1-score (%)	Precision (%)
1	Relative Strenght Index (RSI)	31.78%	--	--	--	--
2	Lineer Regression (LR)	40.62 %	40.87%	80.42%	42.17%	43.55%
3	Bagged Trees Regressor(BAG)	42.96%	43.24%	81.13%	50.28%	46.97%
4	K-Nearest Neighbors (KNN)	43.95%	36.92%	80.61%	39.05%	41.71%
5	Gaussian Naive Bayes (NB)	48.64%	46.12%	82.71%	46.09%	46.06%
6	CNN-TA	48.87%	45.17%	82.46%	45.43%	45.69%
7	Multi-Layer Perceptron (MLP)	50.51%	46.91%	83.05%	47.35%	47.81%
8	Random Forests (RF)	55.72%	51.60%	84.58%	53.02%	54.53%
9	*CNN-DBEMA*	**57.28%**	**57.43%**	**87.04%**	**56.53%**	**55.67%**
10	Average (%)	46.70%	45.65%	82.75%	47.30%	47.74%

Table 4.1, Table 4.2, and Table 4.3 includes performance evaluation of the models compared by using the datasets with DBEMA features. The next step of our study is to compare the models using datasets without DBEMA features for revealing the importance of DBEMA features. Table 4.4, Table 4.5, and Table 4.6 present the experimental results of the models when datasets without DBEMA features are used for training and testing the models. As it can be easily seen from Table 4.4, 4.5, and 4.6 predictive performance of the models reduce significantly when DBEMA features are not used. DBEMA features significantly change the performance of the benchmark models especially our proposed model. Table 4.8 denotes performance of prediction accuracy of our proposed model. The accuracy score increases from 48.29% to 57.28% with the inclusion of the DBEMA features. This remarkable rise demonstrates significance of DBEMA feature and CNN-DBEMA.

Table 4.4. Performance Evaluation of Proposed Model (CNN-DBEMA) versus Benchmarking Methods with Cross-Over (Last Day) Points of Data without DBEMA features.

No	**Method Name**	Acc_Scr (%)	Se (%)	Sp (%)	F-score (%)	Precision(%)
1	Relative Strenght Index(RSI)	31.78%	--	-	--	--
2	Lineer Regression (LR)	35.95%	36.84%	78.86%	41.77%	46.42%
3	Bagged Trees Regressor(BAG)	39.07%	39.05%	79.79%	43.76%	49.78%
4	CNN-TA	43.95%	36.95%	80.69%	39.05%	41.71%
5	K-Nearest Neighbors (KNN)	48.09%	44.72%	82.16%	44.98%	45.26%
6	CNN-DBEMA	48.29%	40.98%	81.67%	42.65%	44.48%
7	Gaussian Naive Bayes (NB)	51.36%	49.12%	83.53%	49.20%	49.30%
8	Random Forests (RF)	52.45%	48.88%	83.60%	**53.86%**	49.98%
9	Multi-Layer Perceptron (MLP)	**52.76%**	**50.23%**	**83.90%**	50.50%	**50.77%**
10	Average (%)	44.85%	43.34%	81.77%	45.72%	47.21%

Table 4.5. Performance Evaluation of Proposed Model (CNN-DBEMA) versus Benchmarking Methods with All of 75 days points of Data without DBEMA features.

No	**Method Name**	Acc_Scr (%)	Se (%)	Sp (%)	F-score (%)	Precision(%)
1	Relative Strenght Index(RSI)	31.78%	--	-	--	--
2	Lineer Regression (LR)	35.88%	35.19%	78.73%	36.79%	38.55%
3	Bagged Trees Regressor(BAG)	39.92%	39.71%	80.03%	43.20%	47.38%
4	CNN-TA	43.95%	36.95%	80.69%	39.05%	41.71%
5	K-Nearest Neighbors (KNN)	47.70%	43.21%	81.91%	43.04%	42.88%
6	CNN-DBEMA	48.29%	40.98%	81.67%	42.65%	44.48%
7	Multi-Layer Perceptron (MLP)	50.51%	48.04%	83.20%	48.07%	48.11%
8	Gaussian Naive Bayes (NB)	50.89%	47.53%	83.30%	47.58%	47.63%
9	Random Forests (RF)	**54.71%**	**50.22%**	**84.47%**	**51.19%**	**52.21%**
10	Average (%)	44.84%	42.72%	81.75%	43.94%	45.36%

Table 4.6. Performance Evaluation of Proposed Model (CNN-DBEMA) versus Benchmarking Methods with Average Points of Data without DBEMA features.

No	**Method Name**	Acc_Scr (%)	Se (%)	Sp (%)	F-score (%)	Precision(%)
1	Bagged Trees Regressor(BAG)	25.45%	24.34%	75.05%	46.53%	31.99%
2	K-Nearest Neighbors (KNN)	28.33%	23.63%	75.33%	24.17%	24.75%
3	Lineer Regression (LR)	29.03%	27.43%	76.20%	31.39%	38.21%
4	Relative Strenght Index(RSI)	31.78%	--	-	--	--
5	Random Forests (RF)	34.79%	29.64%	77.29%	28.26%	27.01%
6	Gaussian Naive Bayes (NB)	36.19%	32.64%	78.25%	32.65%	32.67%
7	Multi-Layer Perceptron (MLP)	37.43%	32.93%	78.38%	31.99%	31.12%
8	CNN-TA	43.95%	36.95%	80.69%	39.05%	41.71%
6	CNN-DBEMA	**48.29%**	**40.98%**	**81.67%**	**42.65%**	**44.48%**
10	Average (%)	35.02%	31.07%	77.86%	34.59%	33.99%

Table 4.7. Effect of DBEMA Features on Benchmark Models

No	**Metric**	**With DBEMA**	**Without DBEMA**	**Effect of DBEMA**	**Data Type**
1	Average Accuracy	47.47%	44.85%	+2.62 %	Crossover Point
2	Average Accuracy	41.19%	35.02%	+6.17 %	Average Point
3	*Average Accuracy*	*46.70%*	*44.84%*	*+1.86 %*	*All Point*

Table 4.8. Effect of DBEMA Features on Proposed Model (CNN-DBEMA)

No	**Metric**	**With DBEMA**	**Without DBEMA**	**Effect of DBEMA**
1	Accuracy	57.28%	48.29%	+8.99 %

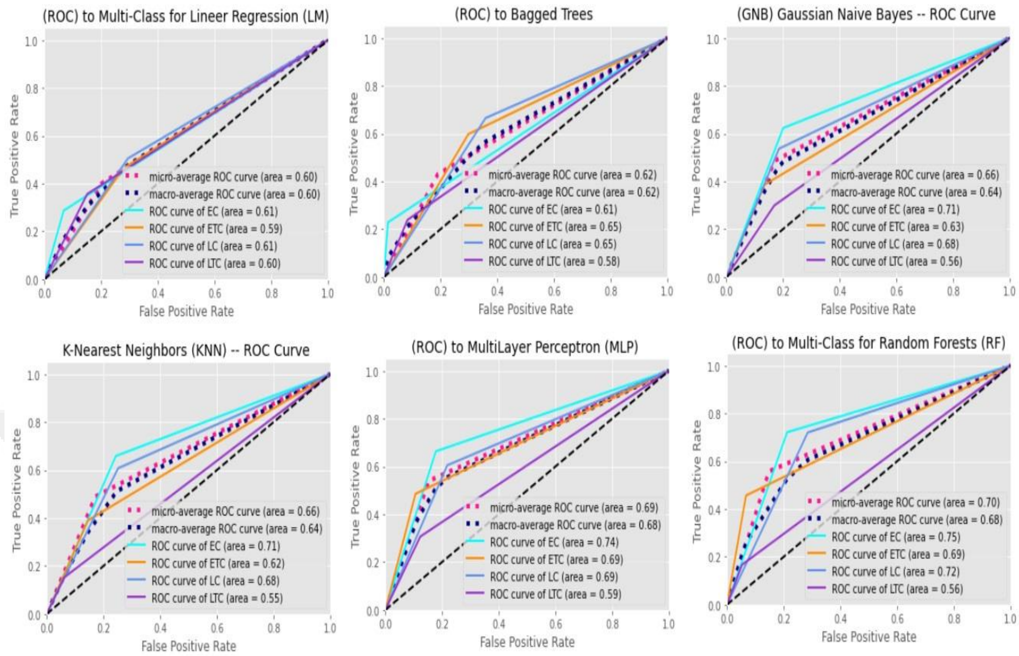


Figure 4.1. Multi-class result ROC curves for all 75 days of data.

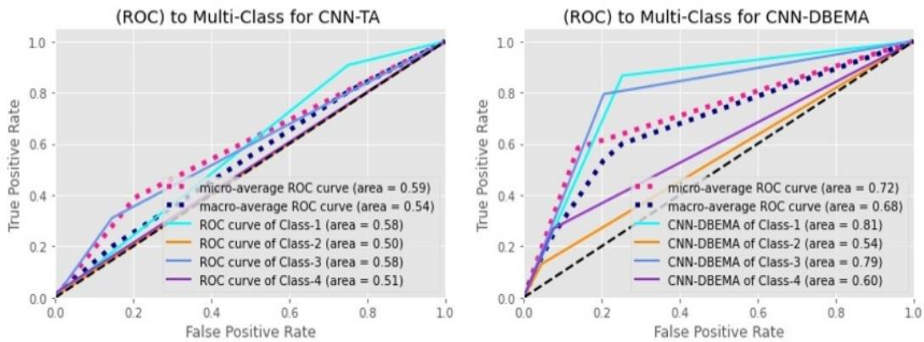


Figure 4.2. Multi-class result ROC curves for CNN-TA and CNN-DBEMA.

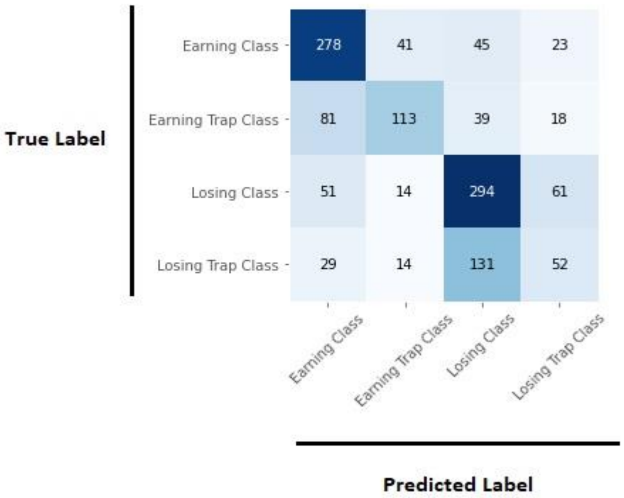


Figure 4.3. Confusion matrix for CNN model with DBEMA

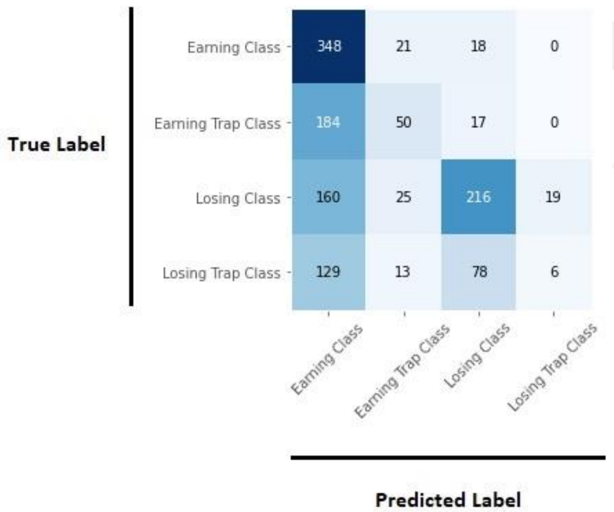


Figure 4.4. Confusion matrix for CNN model without DBEMA

Consequently, there are limited number of CNN-based applications on financial time series prediction in literature. The performance of our proposed model was compared with similar algorithms implemented in the following studies.

- (Hoseinzade and Haratizadeh 2019) have proposed 2D-CNN-based framework to predict next day value of five different index which are NASDAQ Composite, S&P 500 index, NYSE Composite, Dow Jones Industrial Average and, RUSSELL 2000 with F-Measure metric. The average F-Measure scores are 49,44%, 49.14%, 48.85%, 49.75%, and 50.02% respectively.
- (Tekin and Çanakoğlu 2018) were examined 25 Bist companies and LR, MLP, RF, NB, and SVM models were used to predict next day daily return rate with using previous 20 days return values. Three different datasets were used. The average accuracy score is 55.62%.
- (Gündüz et al. 2018) forecasted the direction movement of the nine banking stocks in Bist using LSTM. The Naive and Random methods were compared with LSTM. The hourly data was used. The experimental result was evaluated by the macro averaged F-Score. The F-Score rate is 54.0%.
- (Chen et al. 2021) designed a novel method that is CNN-based graph convolutional feature for trend prediction of stocks using both stock market information and correlation between individual stock information. Six Chinese stocks are used to present the performance of the GC–CNN based model. The best accuracy score is 53.37% and average annual return is 33.42%.
- (Sezer and Ozbayoglu 2018) proposed a novel method CNN-TA to predict daily stock price. The new model used 2D imaged-like data and each image was created using 15 different financial technical indicators and 15 days period. The system was trained on CNN model and each imaged-like data was labeled as Sell, Buy, and Hold. The experimental results show that prediction accuracy of Dow-30 is 58% and maximum annual return is 31.64% with AXP stock.

- (Gunduz et al. 2017) have designed a CNN model with organized features to predict the hourly intraday movement of Bist 100 stock. The proposed model was compared randomly ordered CNN and Logistic Regression. The proposed CNN model outperforms better prediction result. The proposed CNN had 0.563 F-score rate with the reordered attributes.

When we compare our proposed method with similar studies in literature, it is understood that our proposed methods has better accuracy score. (Gunduz et al. 2017) have purposed a CNN model with organized features has achieved very close results. There are limited studies that show the financial results. We compare our proposed model's financial results with (Sezer and Ozbayoglu 2018)'s study, it is understood that our proposed model get better financial success.

4.3. Financial performance evaluation of CNN-DBEMA

In this section, the financial performance of the proposed CNN-DBEMA model was evaluated, and the system adopted stocks daily from the BIST. The prediction strategy for our proposed model is indicated in Algorithm 3, Fig. 4.5, and described in detail in section 3.3. The BIST data covered from June 1, 1998, to March 31, 2021, for a total of 5672 trading days. BIST stock data from June 1, 1998, to July 24, 2017, was used for training, and the data from July 24, 2017, to March 31, 2021, was used for testing procedures. At this point, the proposed algorithmic trading model evaluated the transactions, which were formed in an alert system list. Every day, the system checked the relationship between the EMA9 and EMA200. If any of the scanned stock's EMA9 values crossed over the EMA200 value, the system generated the stock's name and date in the alert system list. Using this approach, stocks were saved in the alert list displayed in Table 4.9. By monitoring the alert system list, the system created investment transactions for the adopted stock. If there was

Table 4.9. Demonstration of Alert System (First Ten Records)

No	Alert Date	Stock Code
1	26.07.2017	KONYA
2	28.07.2017	NTHOL
3	31.07.2017	AKCNS
4	11.08.2017	KENT
5	23.08.2017	BRISA
6	24.08.2017	GOLTS
7	11.09.2017	METUR
8	13.09.2017	OYAKC
9	14.09.2017	CIMSA
10	15.09.2017	KENT

Table 4.10. Turkish National Bank Interest Rates (Annual)

No	Date	Interest Rate(IR)	Average -IR(%)	Years Order	Capital Amount
0	25-01-2017	--	--	--	100 TL
1	25-01-2017	7,25	7,25	1.Year	107.25 T
2	01-06-2018	15,00			
3	08-06-2018	6,25	19,06	2.Year	127.69 TL
4	14-09-2018	2,50			
5	21-09-2018	22,50			
6	26-07-2019	18,25			
7	13-09-2019	15,00	4,06	3.Year	145.65 TL
8	25-10-2019	12,50			
9	13-12-2019	10,50			
10	17-01-2020	9,75			
11	20-02-2020	9,25			
12	18-03-2020	8,25			
13	23-04-2020	7,25			
14	22-05-2020	6,75	0,28	4.Year	160.62 TL
15	25-09-2020	8,75			
16	20-11-2020	13,50			
17	27-11-2020	13,50			
18	25-12-2020	15,50			
19	19-03-2021	17,50			
20	24-09-2021	16,50			
21	22-10-2021	14,50	4,90	5.Year	84.55 TL
22	19-11-2021	13,50			

More than one stock on the alert system list on the same day, the system selected the stock with the highest prediction score. The other control mechanism started to check the prediction score. If the prediction score is more than the threshold probability score, the system bought the stock and a holding phase was triggered. If the price of the selected stock jumped over the threshold return ratio (10%), the system would sell the stock and close the transaction. Then the system would check the alert system list again, after the previous stock's closed transaction date. If the selected stock did not pass the threshold return ratio amount, the system would sell the stock at the end of the transaction waiting time. The waiting time was defined as 60 days, as explained in section 3.3. The pseudocode of the financial method is expressed in Algorithm 3.

The financial assessment process is denoted in Table 4.11, where we started our investment with 100 TL capital on the date 04-10-2017 and invested in stock BRYAT for 57 days. On 30.11.2017 system sold the BRYAT stocks and increased capital to 110 TL. Our system gave a buy signal on 01.12.2017 for BRYAT stock again, we invested in that stock with 110 TL capital. 13 days later, our system reach the sell signal ratio and sold automatically BRYAT stocks, then our capital increased to 121 TL. As shown in Table 4.11, at the end of the financial evaluation, on day 08.11.2019 our proposed model increased the capital from 100 TL to 1003 TL (within two years). To compare the proposed CNN-DBEMA model with the Turkish Republic National Bank interest ratio (annual), Table 4.10 denotes what happens when we preferred bank interest rates for investment. In a five-year period, 100 TL of capital was raised to 184 TL with respect to National Bank interest rates. When Table 4.10 and Table 4.11 are compared, it can be seen that the proposed model (CNN-DBEMA) demonstrated superior financial performance.

Algorithm 3: Decide to invest capital on stock

```
    Set with_in_day to 60                          // Max 60 days later
    Set Earn_ratio to 10                           // Min %10 profit
    Set Accuracy to 0.75                           // Prediction success of threshold
    Set Control to False                           // Limit of end date
    Set Alert_system_list to []                    // Alert for invest to stock list
    Set Feature_list to []                         // Stock feature list
    Set Call_back_days to 75                       // Evaluate stock last 75 days
    Set Alert_date                                 // Alert day of stock
    Set Enter_transaction to False                 // Decide to invest parameter
    Set Earn to 0                                  // Earn class label
    Set Earn_Ratio to 10                           // Threshold of Earn Ratio
    Set End_day                                    // Set End day for control transactions
WHILE Control:
    IF  lenght(Alert_system_list) > 0
        alert_date to Alert_system_list["Index"] and capital to 100
        FOR stock_i to Call Alert_system_list(Start_date):
            Call load_stock_datas(stock_i ):
            stock_i = stock_data[Feature_list] between alert_date-Call_back_days and alert_date
            Call MinMaxScaler_Function(stock_i) to img
            Call model.predict(img) to Model_array_result
            FOR index=0 to Call range(0,lenght(Model_array_result)):
                    IF Model_array_result["Index"] THEN
                        ADD to Alert_result
        FOR index =0 to Call range(0,lenght(Alert_result)):
            IF Alert_result["class"] ["Index"] = Earn and Alert_result["Score"] ["Index"] > Accuracy THEN
                enter_transaction = True
        IF enter_transaction:
            CALL get_highest_prediction_score(Alert_result) to selected_stock
                IF lenght(Alert_result) >0 THEN
                    CALL load_stock_datas(selected_stock ):
                    CALL update(index):
                CALL load_return_ratio (selected_stock ) to earn_ratio
                IF  earn_ratio  > Earn_Ratio THEN
                    capital ← ( capital  +  capital* Earn_Ratio)
                    Index ←  CALL next( Index) and  alert_date ← Alert_system_list["Index"]
                ELSE
                    Index ←  CALL next( Index) and  alert_date ← Alert_system_list["Index"]
                    capital ← CALL next_Index_Value(capital)
                IF   alert_date > End_day  THEN
                    Control ← False
                ELSE
                    Control ← True  and  CALL update(Alert_system_list["Index"])
        ELSE
            Control ← True  and CALL update(Alert_system_list["Index"])
    ELSE
    Control ← False
```

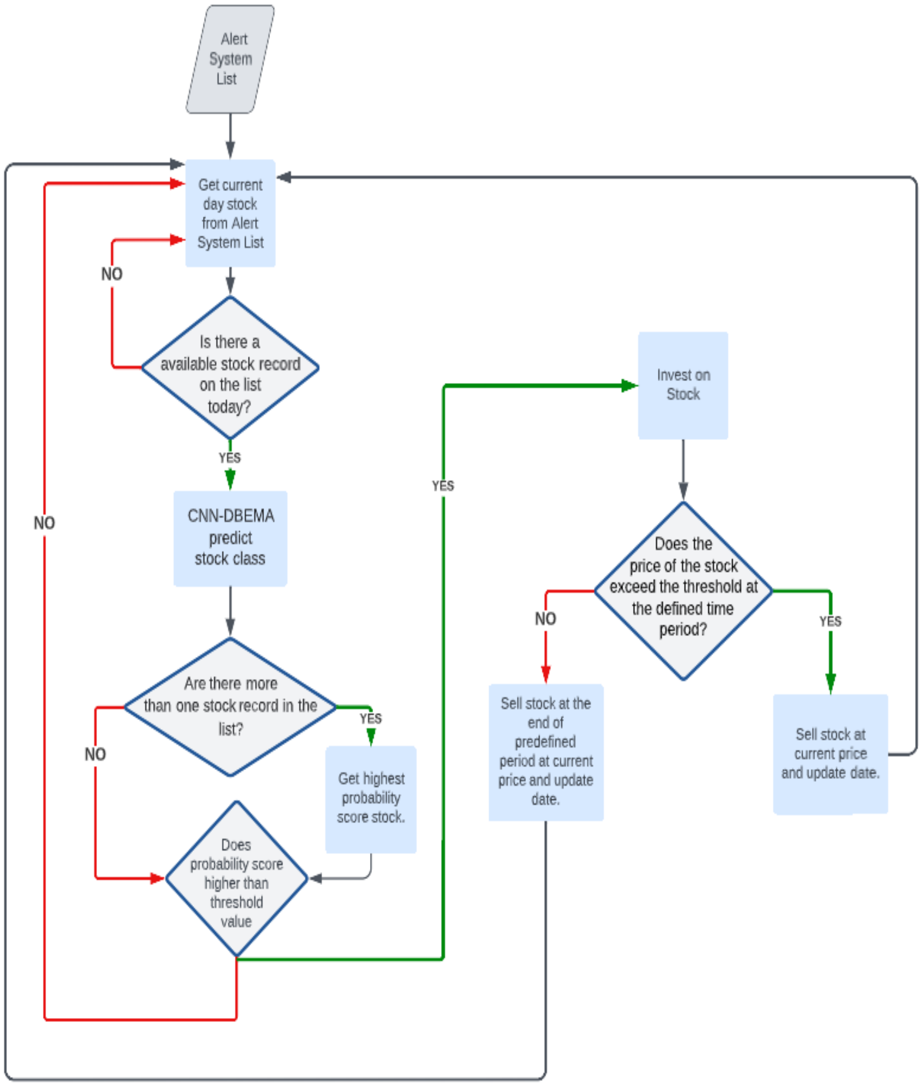


Figure 4.5. Flow Chart of Investment Decision on Stock

Table 4. 11. Financial Performance Evaluation of The Proposed Method CNN-Dbema

No	Start Date	End Date	Latency	Predicted Stock	Result	Capital
0	04.10.2017	--	--	--	--	100 TL
1	04.10.2017	30.11.2017	57 Days	BRYAT	Successful	110 TL
2	01.12.2017	14.12.2017	13 Days	BRYAT	Successful	121 TL
3	15.12.2017	10.01.2018	26 Days	BRISA	Successful	133 TL
4	18.01.2018	24.01.2018	6 Days	PEGYO	Successful	146 TL
5	29.01.2018	21.02.2018	23 Days	OZGYO	Successful	161 TL
6	25.04.2018	25.06.2018	60 Days	EGPRO	Failed	160 TL
7	27.06.2018	02.07.2018	5 Days	KRDMB	Successful	176 TL
8	06.07.2018	19.07.2018	13 Days	EGEEN	Successful	194 TL
9	23.07.2018	24.07.2018	2 Days	DERIM	Successful	213 TL
10	25.07.2018	14.08.2018	20 Days	KARTN	Successful	234 TL
11	27.08.2018	31.08.2018	4 Days	TIRE	Successful	258 TL
12	07.09.2018	20.09.2018	13 Days	KCHOL	Successful	284 TL
13	12.10.2018	11.12.2018	60 Days	DERIM	Failed	264 TL
14	13.12.2018	11.02.2019	60 Days	ATEKS	Failed	264 TL
15	15.02.2019	18.02.2019	3 Days	AVGYO	Successful	290 TL
16	26.02.2019	27.02.2019	2 Days	VESTL	Successful	319 TL
17	01.03.2019	18.03.2019	17 Days	BRSAN	Successful	351 TL
18	20.03.2019	30.04.2019	41 Days	VAKKO	Successful	386 TL
19	03.05.2019	07.05.2019	4 Days	DGGYO	Successful	425 TL
20	21.05.2019	28.05.2019	7 Days	BANVT	Successful	467 TL
21	30.05.2019	18.06.2019	19 Days	FROTO	Successful	514 TL
22	19.06.2019	21.06.2019	2 Days	SONME	Successful	566 TL
23	24.06.2019	15.08.2019	51 Days	SNPAM	Successful	622 TL
24	16.08.2019	23.08.2019	7 Days	PEGYO	Successful	685 TL
25	29.08.2019	05.09.2019	7 Days	ADEL	Successful	753 TL
26	06.09.2019	09.09.2019	3 Days	PNSUT	Successful	829 TL
27	10.09.2019	30.09.2019	20 Days	KRSTL	Successful	912 TL
28	10.10.2019	08.11.2019	29 Days	ICBCT	Successful	1003 TL

5. CONCLUSIONS

In this thesis, we proposed to design a novel CNN-DBEMA method for time series prediction especially financial time series by considering both the new concept of DBEMA and technical indicators. In the literature, mostly three types of classification have been used, buy, sell and hold (Namir et al., 2022; Brabazon et al., 2020). The novel proposed DBEMA method was devised to predict four classes—earning class (EC), earning trap class (ETC), losing class (LC), and losing trap class (LTC)—using a CNN. The most important innovation in this study is the discovery that the distances between the exponential moving averages produce predictions for financial time series. At the same time, these new distance-based features can be applied as new parameters in other models and will increase the success of estimation.

For increasing the efficiency of the system, a hybrid system has been proposed. The most suitable technical indicators were determined with the two wrapper based feature selection methods. The first wrapper method for adopting suitable feature subsets was the recursive feature elimination (RFE) algorithm using classification and regression tree (CART) and the second method was the sequential feature selection (SFS) algorithm using gradient boosting algorithm (GBM) to regularize classification penalty. The reason for selecting wrapper algorithms is to achieve more accurate results and adopting GBM and SFS is to be more suitable for regression and classification problems. Then, the combination of the selected technical indicators and the proposed DBEMA features are sent to CNN to determine class label.

Most of the CNN based studies in the literature were performed for computer vision and image processing (Karpathy, 2014). Recently, image-based financial series prediction has become a significant issue. In our study, we created 6425 images with 18 technical indicators and 6 DBEMA parameters using the end-of-day values of 118 stocks in the BIST from 1998 to 2021. Experimental results were obtained for 80% of the dataset as a training set, which involved data from 1998 to 2017, and 20% of the dataset as a testing set, which contained data from 2017 to 2021. Each image involved data for 75 working days of stock. The length of the period was determined based on our experimental results.

In the comparison phase, we selected well-known benchmarking methods, including regression models, financial methods, classification models such as linear regressions (LR), random forests (RF), multi-layer perceptron (MLP), bagged trees regressors (BAG), a novel CNN-TA, and the relative strength index method (RSI). The CNN-DBEMA model got the highest accuracy score among the selected benchmarking methods. The CNN-DBEMA model outperformed the best machine-learning method, with an accuracy score of 57.28%, as seen in Table 4.2 and Table 4.3. Financial evaluation of the proposed model also proved the success of the CNN-DBEMA model. CNN has an advantage in terms of data integrity when used in financial time series. In other models, there is a challenge to use the data as average, endpoint (cross-over), or all points. In this respect, we tried to prove to the researchers that time-series estimation with CNN is more efficient.

Though the prediction capability of machine-learning methodologies in flexible-market prediction was established, it still has room for improvement.

REFERENCES

Alzubaidi, L., Al-Shamma, O., M. Fadhel, A., Farhan, L., Zhang, J, and Duan, Y., 2020. “Optimizing the Performance of Breast Cancer Classification by Employing the Same Domain Transfer Learning from Hybrid Deep Convolutional Neural Network Model,” Electronics, vol. 9, no. 3, Art. no. 445, DOI: 10.3390/electronics9030445.

Alzubaidi, L., et al., 2020. “Towards a Better Understanding of Transfer Learning for Medical Imaging: A Case Study,” Appl. Sci, vol. 10, no. 13, Art. no. 4523, DOI: 10.3390/app10134523Uthaib,

Ananthi, M. Vijayakumar, K. 2020 Stock market analysis using candlestick regression and market trend prediction (CKRM), J. Ambient Intell. Humanize. Computation. , https://doi.org/10.1007/s12652-020-01892-5.

Aydin, A., D., Cavdar, S., C., 2015 "Comparison of Prediction Performances of Artificial Neural Network (ANN) and Vector Autoregressive (VAR) Models by Using the Macroeconomic Variables of Gold Prices, Borsa Istanbul (BIST) 100 Index and US Dollar-Turkish Lira (USD/TRY) Exchange Rates." Procedia Economics and Finance Volume 30, 2015, Pages 3-14 https://doi.org/10.1016/S2212-5671(15)01249-6

Aycel, U., Santur Y., 2022 "A new moving average approach to predict the direction of stock movements in algorithmic trading.", Journal of New Results in Science 11(1) (2022) 13-25 https://doi.org/10.54187/jnrs.979836

Ballings, M., Poel, D. V. d. Hespeels, N. Gryp, R. 2015 Evaluating multiple classifiers for stock price direction prediction, Expert Syst. Appl. 42 (20) 7046–7056.

Brabazon, A., Kampouridis, M. and O’Neill, M., (2020). Applications of genetic programming to finance and economics: past, present, future. Genetic Programming and Evolvable Machines. 21 (1-2), 33-53

Benny, R., Ian, B., 2022 "Autoregressive mixture models for clustering time series." Journal of Time Series Analysis. https://doi.org/10.1111/jtsa.12644

Bildirici, M., Ersin, Ö., Ö., 2009 "Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul Stock Exchange." Expert Systems with Applications Volume 36, Issue 4, May 2009, Pages 7355-7362 https://doi.org/10.1016/j.eswa.2008.09.051

Box, G., E., Jenkins, G., M., Reinsel, G., C., Ljung, G., M., 2015 Time Series Analysis: Forecasting and Control, John Wiley & Sons,

Box, G.E.P., Jenkins, G. M., 1994 Reinsel, G. C. Time series analysis: Forecasting and control (3rd ed.), Prentice Hall, New Jersey

Boyacioglu, M., A., Avci, D., 2010 "An Adaptive Network-Based Fuzzy Inference System (ANFIS) for the prediction of stock market return: The case of the Istanbul Stock Exchange." Expert Systems with Applications Volume 37, Issue 12, December 2010, Pages 7908-7912 https://doi.org/10.1016/j.eswa.2010.04.045

Cao, L., J., Tay, F.E.H., 2003 "Support vector machine with adaptive parameters in financial time series forecasting." IEEE Transactions on Neural Networks Volume: 14, Issue: 6 DOI: 10.1109/TNN.2003.820556

Cagliero, L. Garza, P. Attanasio, G. Baralis, E. 2020 Training ensembles of faceted classification models for quantitative stock trading, Computing 102 (5) 1213–1225.

Chang, V., Li, T., Zeng, Z., 2019. "Towards an improved Adaboost algorithmic method for computational financial analysis". Journal of Parallel and Distributed Computing Volume 134, December 2019, Pages 219-232. https://doi.org/10.1016/j.jpdc.2019.07.014

Chen, M., Y., Liao, C., H., Hsieh, R., P., 2019 "Modeling public mood and emotion: Stock market trend prediction with anticipatory computing approach.", Comput. Hum. Behav. 101 402–408, https://doi.org/10.1016/j.chb.2019.03.021

Chen, W. Jiang, M. Zhang, W.G. Chen, Z. 2021 A novel graph convolutional feature-based convolutional neural network for stock trend prediction. Information Sciences Volume 556, Pages 67-94. https://doi.org/10.1016/j.ins.2020.12.068.

Chen, M.-Y. Chen, Chen, B.-T. 2015 A hybrid fuzzy time series model based on granular computing for stock price forecasting, Inf. Sci. 294 227–241.

Chong, E., Han, C., Park, F., C., 2017 "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies." Expert Systems With Applications 83 187–205 https://doi.org/10.1016/j.eswa.2017.04.030

Dasha, R., KishoreDash, P., 2016. "A hybrid stock trading framework integrating technical analysis with machine learning techniques". The Journal of Finance and Data Science. Volume 2, Issue 1, March 2016, Pages 42-57 https://doi.org/10.1016/j.jfds.2016.03.002

Dey, P. P. Nahar, N. 2020 Forecasting Stock Market Trend using Machine Learning Algorithms with Technical Indicators. I.J. Information Technology and Computer Science, 3, 32-38. https://doi.org/10.5815/ijitcs.2020.03.05

Dietterich, T. G. 2000. Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1–15). Springer.

Durham, G. B., 2007 Sv mixture models with application to s&p 500 index returns, Journal of Financial Economics 85 (3) 822–856.

Ferri, F. J., Pudil P., Hatef, M., Kittler, J. (1994). "Comparative study of techniques for large-scale feature selection." Pattern Recognition in Practice IV : 403-413.

Guancen, L., Aijing, L., Jianing, C., 2021 "Multidimensional KNN algorithm based on EEMD and complexity measures in financial time series forecasting."

Expert Systems with Applications Volume 168, 15 April 2021, https://doi.org/10.1016/j.eswa.2020.114443

Gündüz, H., Yaslan, Y., Cataltepe, Z., 2018 "Stock market prediction with deep learning using financial news." DOI: 10.1109/SIU.2018.8404616 Conference: 2018 26th Signal Processing and Communications Applications Conference (SIU)

Gunduz, H., Yaslan, Y., Cataltepe, Z., 2017. "Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations" Knowledge-Based Systems 137 138–148. https://doi.org/10.1016/j.knosys.2017.09.023.

Guresen, E. Kayakutlu, G. Daim, T.U. 2011 Using artificial neural network models in stock market index prediction, Expert Syst. Appl. 38 (8) 10389–10397.

Hasan, A., Kalıpsız, O., Akyokus, S., 2018 "Application of Machine Learning Techniques for the Prediction of Financial Market Direction on BIST 100 Index." COMPUTATIONAL METHODS AND TELECOMMUNICATION IN ELECTRICAL ENGINEERING AND FINANCE May 6-9, 2018

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 (pp. 770–778).

Hoseinzade, E., Haratizadeh. S., 2019 CNNpred: CNN-base d stock market prediction using a diverse set of variables Expert Systems With Applications https://doi.org/10.48550/arXiv.1810.08923

Huang, C., 2012. "A hybrid stock selection model using genetic algorithms and support vector regression". In: Applied Soft Computing 12.2, pp. 807–818. https://doi.org/10.1016/j.asoc.2011.10.009

Huy, T., P., Cuong, N., T., 2018 "Effectiveness of Investment Strategies Based on Technical Indicators: Evidence from Vietnamese Stock Markets." Journal of Insurance and Financial Management Vol 3, No 5.

Ihab A. El-Khodary, Ahmed Zeweil St. 2009 A Decision Support System for Technical Analysis of Financial Markets Based on the Moving Average Crossover World Applied Sciences Journal, 6, 1457-1472.

Jiang, M. Jia, L. Chen, Z. Chen, W. 2020 The two-stage machine learning ensemble models for stock price prediction by combining mode decomposition, extreme learning machine and improved harmony search algorithm, Ann. Oper. Res. https://doi.org/10.1007/s10479-020-03690-w.

Jiang, Z., Ji, R., Chang, K-C., 2020. "A Machine Learning Integrated Portfolio Rebalance Framework with Risk-Aversion Adjustment". J. Risk Financial Manag. 2020, 13(7), 155; https://doi.org/10.3390/jrfm13070155.

Kamalov, F., Gurrib, I. & Rajab, K. 2021. Financial Forecasting with Machine Learning: Price Vs Return. Journal of Computer Science, 17(3), 251-264. https://doi.org/10.3844/jcssp.2021.251.264

Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, 2014 "Large-scale video classification with convolutional neural networks". Proceedings of the IEEE Conference on Computer Vision and pattern recognition 1725–1732.

Kayal, A. 2010 A neural networks filtering mechanism for foreign exchange trading signals, in Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), 3, IEEE, pp. 159–167.

Kim, K. j., 2003 "Financial time series forecasting using support vector machines." Neurocomputing Volume 55, Issues 1–2, September 2003, Pages 307-319

Kini, B.V. Sekhar, C.C. 2013 Large margin mixture of ar models for time series classification, Appl. Soft Comput. 13 361–371.

Kirisci, M., Yolcu, O, C., 2022. "A New CNN-Based Model for Financial Time Series: TAIEX and FTSE Stocks Forecasting" Neural Processing Letters https://doi.org/10.1007/s11063-022-10767-z

Kraussa, C., AnhDoa, X., Huck, N., (2017). “Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500”. In: European Journal of Operational Research 259.2, pp. 689–702.

Krizhevsky, A., I. Sutskever, G.E. Hinton, 2012, "Imagenet classification with deep convolutional neural networks." , Adv. Neural Inform. Process. Syst. 1097–1105.

Lawrence, S., Giles,C.L. A.C. Tsoi, A.D. Back, 1997 "Face recognition: a convolutionalneural-network approach." , IEEE Trans. Neural Netw. 8 (1) 98–113.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521 (7553), 436.

Lei, Y., Peng, Q., Shen, Y., 2020. "Deep Learning for Algorithmic Trading: Enhancing MACD Strategy". ICCAI '20: Proceedings of the 2020 6th International Conference on Computing and Artificial IntelligenceApril 2020 Pages 51–57 https://doi.org/10.1145/3404555.3404604

Levantesi, S., Zacchia, G., 2021 "Machine Learning and Financial Literacy: An Exploration of Factors Influencing Financial Knowledge in Italy Journal of Risk and Financial Management". Journal of Risk and Financial Management 14: 120 https://doi.org/10.3390/jrfm14030120

Liang, Q., Rong, W., Zhang, J., Liu, J., Xiong, Z., 2017 "Restricted boltzmann machine based stock market trend prediction.", International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1380–1387. DOI: 10.1109/IJCNN.2017.7966014

Liu, S., Zhang, C., Ma. J., 2017 "Cnn-lstm neural network model for quantitative strategy analysis in stock markets." Neural Information Processing, pages 198–206. Springer International

Luenberger, D. 1979 Introduction to Dynamic Systems: Theory, Models, and Applications, Wiley.

Lütkepohl, H. 2005 New Introduction to Multiple Time Series Analysis, Springer-Verlag.

M.A., Croock, M. S., 2021 “Multi classification of license plate based on deep convolution neural networks.” International Journal of Electrical and Computer Engineering. pp. 5266~5276 ISSN: 2088-8708, DOI: 10.11591/ijece.v11i6.pp5266-5276

Ma, Z., Dai, Q., Liu, N., 2015 "Several novel evaluation measures for rank-based ensemble pruning with applications to time series prediction." Expert Systems with Applications, 42 (1) , pp. 280-292

Mabu, S. Obayashi, M. Kuramoto, T. 2015 Ensemble learning of rule-based evolutionary algorithm using multi-layer perceptron for supporting decisions in stock trading problems, Appl. Soft Comput. 36 357–367.

Markovic, I.P. Stojanovic, M.B. Stankovic, J.Z. Božic, M.M. 2014 Stock market trend prediction using support vector machines, Facta Univ. Ser. Autom. Control Robot. 13 (3) 147–158 http://casopisi.junis.ni.ac.rs/index.php/FUAutContRob/article/view/585

Menkhoff, L., Taylor, M. P. 2007. The obstinate passion of foreign exchange professionals: Technical analysis. Journal of Economic Literature, 45 December 4, 936–972

Moews, B., Herrmann, J., M., Ibikunle, G., "Lagged correlation-based deep learning for directional trend change prediction in financial time series.", Expert Syst. Appl. 120 (2019) 197–206. https://doi.org/10.1016/j.eswa.2018.11.027

Naik, N., Mohan, B., R., 2020 "Intraday stock prediction based on deep neural network." National Academy Science Letters 43 241–246.

Namir, K., Labriji, H., Lahmara, E., H., B., 2022 "Decision Support Tool for Dynamic Inventory Management using Machine Learning, Time Series and Combinatorial Optimization". Procedia Computer Science Volume 198, 2022, Pages 423-428. https://doi.org/10.1016/j.procs.2021.12.264

Nassirtoussi, A.K., Aghabozorgi, S., Wah, T.Y. ve Ngo, D.C.L. 2014. Text mining for market prediction: A systematic review, Expert Systems with Applications, 41(16), 7653–7670.

Nelson, D., M., Q., Pereira, A., C., M., Oliveira, R., A., 2017 "Stock market's price movement prediction with LSTM neural networks". 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1419–1426. DOI: 10.1109/IJCNN.2017.7966019

Ntakaris, A., Kanniainen, J., Gabbouj, M., Iosifidis, A., 2020. "Mid-price prediction based on machine learning methods with technical and quantitative indicators". PLoS One. 2020; 15(6): e0234107 https://doi.org/10.1371/journal.pone.0234107.

O.B. Sezer, A.M. Ozbayoglu, 2018 "Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach", Appl.Soft. Comput. 70 525–538.

O.B. Sezer, A.M. Ozbayoglu, E. Dogdu, 2017 An artificial neural network-based stock trading system using technical analysis and big data framework, in: Proceedings of the SouthEast Conference, ACM, pp. 223–226.

Oriani, F. B. Coelho, G. P. 2017 Evaluating the impact of technical indicators on stock forecasting. IEEE Symposium Series on Computational Intelligence (SSCI). February 13, 2017. https://doi.org/10.1109/SSCI.2016.7850017

Pardeshi, Y., K., Kale, P., 2021 "Technical Analysis Indicators in Stock Market Using Machine Learning: A Comparative Analysis". 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). https://doi.org/10.1109/ICCCNT51525.2021.9580172

Park, C.-H., Irwin, S., H., 2007. What do we know about the profitability of technical analysis Journal of Economic Surveys, 21 September 4 , 786–826.

Patel, J., Shah, S., Thakkar, P. ve Kotecha, K. 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, Expert Systems with Applications, 42(1), 259–268.

Pehlivanlı, A., Ç., Barış Aşıkgil Güzhan Gülay 2016 "Indicator selection with committee decision of filter methods for stock market price trend in ISE." Applied Soft Computing Volume 49, December 2016, Pages 792-800 https://doi.org/10.1016/j.asoc.2016.09.004

Persio Di, L. Honchar, O. 2016 Artificial neural networks architectures for stock price prediction: comparisons and applications, Int. J. Circuits Syst. Signal Process. 10 403–413.

Pramudya, R., Ichsani, S., 2020 "Efficiency of Technical Analysis for the Stock Trading." International Journal of Finance & Banking Studies. IJFBS, VOL 9 NO 1. https://doi.org/10.20525/ijfbs.v9i1.666

Rivero, S., S., Rodríguez, P., N., 2010. Linkages in international stock markets: evidence from a classification procedure, Applied Economics, 42(16), 2081–2089.

Rabiner, L. Juang, B. 1986 An introduction to hidden Markov models, IEEE ASSP Mag. 3 (1) 4–16.

Rodolfo, T., F., N., Jéssica, L., S., Vinicius A., S., Herbert, K., 2017. A literature review of technical analysis on stock markets. The Quarterly Review of Economics and Finance, 611–631.

Rousis, P., Papathanasiou, S., 2018 "Is Technical Analysis Profitable on Athens Stock Exchange?". Mega Journal of Business Research Volume 2018, Article ID 61, 24 Pages

Sapankevych, N.I. Sankar, R. 2009 Time series prediction using support vector machines: a survey, IEEE Comput. Intell. Mag. 4 (2) 24–38.

Tekin, S., Çanakoğlu, E., 2018 "Prediction of stock returns in Istanbul stock exchange using machine learning methods." 2018 26th Signal Processing and Communications Applications Conference (SIU) 10.1109/SIU.2018.8404607

Thomas, F., Christopher. K., (2018). "Deep learning with long short-term memory networks for financial market predictions." European Journal of Operational

Research Volume 270, Issue 2, Pages 654-669 https://doi.org/10.1016/j.ejor.2017.11.054

Vui, C.S. Soon, G.K. On, C.K. Alfred, R. Anthony, P. 2013, A review of stock market pre- diction with artificial neural network (ann), in: Proceedings of the IEEE International Conference on Control System, Computing and Engineering (ICCSCE), IEEE, pp. 477–482.

Yanga, M., Wang, J., 2022 "Adaptability of Financial Time Series Prediction Based on BiLSTM." The 8th International Conference on Information Technology and Quantitative Management. Procedia Computer Science 199 18–25

Yasir, M. N., Croock, M. S., 2020. "Cyber DoS attack-based security simulator for VANET," International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 6, pp. 5832–5843, doi: 10.11591/ijece.v10i6.pp5832-5843

Abu-Mostafa., Y., S., Atiya, A., F., (1996). "Introduction to financial forecasting" Appl Intell 6 (3):205–213. https://doi.org/10.1007/BF00126626

Wang, J. Hou, R. Wang, C. Shen, L. 2016 Improved v-support vector regression model based on variable selection and brainstorm optimization for stock price forecasting, Appl. Soft Comput. 49 164–178.

Wang, S., 2020. "The Prediction of Stock Index Movements Based on Machine Learning". ICCAE 2020: Proceedings of the 2020 12th International Conference on Computer and Automation EngineeringFebruary 2020 Pages 1–6
https://doi.org/10.1145/3384613.3384615

Wang, Y., Liu, L., Wu, C., 2020. "Forecasting commodity prices out-of-sample: Can technical indicators help?". International Journal of Forecasting Volume 36, Issue 2, April–June 2020, Pages 666-683 https://doi.org/10.1016/j.ijforecast.2019.08.004

Zhang, J. Li, L. Chen, W. 2020 Predicting stock price using two-Stage machine learning techniques, Comput. Econ. , https://doi.org/10.1007/s10614-020-10013-5.

Zhao, J., Zeng, D., Liang, S., Kang, H., Liu, Q., 2021 "Prediction model for stock price trend based on recurrent neural network." Journal of Ambient Intelligence and Humanized Computing volume 12, pages745–753, https://doi.org/10.1007/s12652-020-02057-0.

BIOGRAPHY

Uğur Ejder. He received his BSc. from Mathematical Engineering of Yildiz Technical University in 2005, MSc from Computational Science and Engineering in 2011, and from Istanbul Technical University. He had been working in Finans Bank Information Technology as a software developer between 2007 and 2011. He had been working AKBANK Information technology as a software developer between 2011 and 2012 and had been working in Telecom as a software architect between 2011 and 2014. Now, he has been working as Information Technologies Specialist at Adana Alparslan Turkes Science and Technology University since 2014. His research interests include Time Series Analysis, Machine Learning, Convolutional Neural Networks, Knowledge-Based Systems, and Expert Systems.