The Seybold Report

ISSN 1533-9211

DOI 10.17605/OSF.IO/Y49NC

BIG DATA ANALYZING FOR PREDICTING PATIENT FUTURE DISEASE USING NOVEL CGRBFN AND TFTCNN NEURAL METHODS

R.GAYAS AHAMMAD

Research Scholar, Department of Computer Science, Vels Institute of Science, Technology & Advanced Studies (VISTAS), Chennai, Tamilnadu. Email: javagayas@gmail.com

Dr. A.S.ARUNACHALAM

Associate Professor, Department of Computer Science, Vels Institute of Science, Technology & Advanced Studies (VISTAS), Chennai, Tamilnadu. Email: arunachalam1976@gmail.com

Abstract

Electronic health records (EHR) have become more prevalent as a result of the quick development of information technology and Internet technology. This research paper focuses on introducing a novel deep learning approach for classifying the obtained information from feature selection process. The electronic health care records are considered as most important data that has to be protected from anonymous access. The novel CGRBFN model provides an additional cyber security system to product the anonymous access of EHR. The first stage of this research work talks more about removal technique involved for removing irrelevant information from collected EHR and also deals with noise removal techniques and null value entries. The Conjugate Gradient boosting architecture used in the proposed CGRBFN algorithm is chosen based on the attributes or features collected. The cryptographically approach based on block chain technology is used to improve the security of the proposed algorithm. The accuracy, precision, recall and f-measure of the EHR categorization analysis are made. The obtained accuracy for the two dataset MIMIC3 and CDSS are 0.939% and 0.9% respectively. The produced accuracy is more compared with other existing algorithms.

1. INTRODUCTION

The creation of quantitative models for patients that may be used to forecast health status and to aid in the prevention of disease or disability is one of the main objectives of precision medicine. Electronic health records (EHRs) have enormous potential in this regard for quickening clinical research and predictive analysis. Recent research has demonstrated that the secondary use of EHRs has improved patient recruitment for clinical trials, type 2 diabetes subgroup identification, the detection of comorbidity clusters in autistic spectrum disorders, and data-driven drug effects and interaction prediction. Unfortunately, clinical decision support systems or workflows have not reliably incorporated predictive models and tools based on contemporary machine learning techniques.

Due to its high dimensionality, noise, heterogeneity, sparsity, incompleteness, random errors, and systematic biases, EHR data is difficult to represent and model. Even EHR data is most confidential and it is to be protected from anonym's entries. Additionally, several terminologies and codes might be used to express the same clinical profile. For instance, a patient with "type 2 diabetes mellitus" can be recognized by laboratory hemoglobin A1C readings greater than 7.0, the inclusion of the ICD-9 code 250.00, the mention of "type 2 diabetes mellitus" in the free-text clinical notes, and other factors. Machine learning techniques have found it

ISSN 1533-9211

DOI 10.17605/OSF.IO/Y49NC

challenging to recognize patterns that result in predictive clinical models for practical applications.

The selection of features and data representation are key factors in predictive algorithms' success. With EHRs, its usual practice to have a domain expert provide the patterns to look for the learning problem and the objectives to ad hoc specify clinical variables. Even if it is sometimes appropriate, supervised defining of the feature space scales poorly, generalizes poorly, and leaves room for the discovery of new patterns and features. Data-driven strategies for feature selection in EHRs have been proposed to address these drawbacks. The fact that patients are frequently represented as a 2-dimensional vector made up of all the data descriptors existing in the clinical data warehouse is a limitation of these methods.

The choice of features and data presentation are crucial to the effectiveness of proposed prediction algorithms. Having a domain expert provide the patterns to look for the learning problem and targeting to specify clinical variables in an ad hoc manner is a frequent technique used in EHRs. Although it may be useful in some circumstances, the supervised formulation of the feature space scales poorly, generalizes poorly, and leaves room for the discovery of new patterns and features. Data-driven approaches for feature selection in EHRs have been put out as a solution to these drawbacks. These techniques have the drawback that patients are frequently represented as a 2-dimensional vector made up of all the data descriptors existing in the clinical data warehouse.

Due to its sparseness, noise, and repetition, this representation is unsuitable for describing the hierarchical information that is latent or embedded in EHRs. By automatically detecting patterns and dependencies in the data to learn a compact and general representation that makes it easier to automatically extract useful information when building classifiers or other predictors, unsupervised feature learning attempts to overcome limitations of supervised feature space definition. Despite the popularity of deep learning (i.e., learning based on hierarchies of neural networks) and the success of feature learning with text, multimedia, and marketing, these methods have not been widely applied to EHR data. The demonstration of unsupervised deep feature learning is used for pre-process patient-level aggregation in EHR data produces representations that are easier for the computer to understand and dramatically enhances clinical predictive models for a wide range of clinical disorders.

This proposed research work introduces a unique framework for representing patient’s future disease using a set of general attributes that are automatically inferred from a sizable EHR database using proposed deep learning techniques. The representation given to the research work follows framework of "deep patient." In particular, a deep neural network built from a stack of demising auto encoders was employed to process EHRs in an unsupervised manner that recorded consistent patterns and stable data structures that, when combined, make up the deep patient representation.

Deep patient is easy to use for a variety of predicting applications, both supervised and unsupervised, and is domain free (i.e., not connected to any single task because trained over a huge multi-domain dataset). In a large-scale real-world data experiment, researcher can use

The Seybold REPORT

ISSN 1533-9211

DOI 10.17605/OSF.IO/Y49NC

deep patient to predict patients' future diseases and demonstrate that deep patient consistently outperforms both the original EHR representations and common feature learning models. This demonstrates the efficacy of the proposed learning algorithm and accuracy of prediction process.

2. LITERATURE REVIEW

The issue of effectively obtaining phenotypes from longitudinal patient EHRs is known as electronic phenotyping. Hripsak et al. [7] noted that because there are numerous difficulties when dealing directly with raw EHR, this is a crucial step before we can undertake any data-driven applications (such as comparative effectiveness study [13], predictive modelling [14], etc. (Such as the ones we listed in the introduction). The works that have already been done will be outlined below in accordance with the various patient EHR representations.

Representation based on vectors: With this technique, a vector is created for each patient. The value on each dimension represents the summary statistics (e.g., sum, average, max, min, etc.) of the related medical event in a certain time period. Its dimensionality is equal to the number of distinct events that appeared in the EHR. Each phenotype in a vector-based representation which is typically taken to be a linear combination of these unprocessed medical events, and the combination coefficients for generating some sort of optimization approaches [15]. This representation's flaw is that it disregards the temporal connections between such occurrences.

Representation based on tensors: Using this technique, an EHR tensor is created for each patient. Each mode of the tensor denotes a particular kind of medical entity (e.g., patients, medications or diagnosis). The summary co-occurrence statistics of the various occurrences in the appropriate dimensions will be the entry values. A nonnegative tensor factorization-based method for phenotypic extraction from such EHR tensors was suggested by Ho et al.. This approach looked at how various medical entities interacted with one another. The drawback is that they did not yet account for the temporal linkages between events. Representation based on sequences. According to the time stamp of each event, this technique creates an EHR sequence for each patient. The identification of temporal patterns as phenotypes can then be accomplished using common pattern mining techniques [17, 18]. One issue is that this approach typically returns a vast number of patterns due to the considerable diversity among patient EHRs (also known as the "pattern explosion" phenomenon"). Determining which phenotype is clinically beneficial is really tough.

Visualization of Temporal Matrix: With one dimension representing time and the other representing medical events, this method displays the patient EHRs as temporal matrices. A phenotyping strategy was put forth by Zhou et al. [9] by combining medical occurrences with comparable temporal trends. They did not, however, take into account the temporal connections between these events. To find shift-invariant patterns across patient EHR matrices, Wang et al. [11] offered a convolutional matrix factorization approach, but they were unable to identify the ideal pattern lengths and had to enumerate all feasible values. This study proposes an approach that uses temporal matrix representation. Our method can automatically detect significant phenotypes and weigh them in the prediction phase thanks to the clever CNN structure. And

ISSN 1533-9211

DOI 10.17605/OSF.IO/Y49NC

the temporal fusion approach efficiently balances the varied genotypes with varying window widths.

A group of machine learning techniques known as "deep learning" make use of model architectures made up of numerous non-linear transformations in an effort to simulate high-level abstractions in data. Deep learning models have shown outstanding outcomes in computer vision-based technique and speech recognition applications during the last few years. A popular deep learning model is convolutional neural networks (CNN), which is considered as basic model for dividing the model into various levels. CNN is a neural network that uses multiple layers with convolution filters applied to local features and can take advantage of the underlying structure of data (for example, the 2D structure of image data). Each processing unit responds to a tiny portion of input data.

CNN models were initially developed for computer vision, but later it was discovered that they were also useful for retrieving search queries and word embedding learning. Since Collobert et alwork .'s on token-level applications, CNN has been applied to text mining systems for a variety of purposes, including document categorization, sentence modelling, and mining for product features. Only static content can be handled by traditional CNN (e.g., images and documents). Since the patient's condition changes over time, our EHR for the patient is longitudinal. Including the rich temporal data into CNN in order to analyses patient HER will solve the discusses problem. Action recognition and object localization from video sequences are two examples of works that have been presented to capture the temporal information in dynamic circumstances. These techniques typically employ distinct stacks of video frames as input to the network, and they attempt to fuse these stacks of movies using various fusion algorithms on various CNN architectural layers.

The prediction process for upcoming clinical events from historical EHR data, many data sources and modelling techniques have been investigated with considering accuracy as one of the constrains. Krishnan et al. (2013) developed a series of studies using regularized logistic regression to predict diabetes from EHR claims data with varied lengths of the patient history window and prediction window. Tran et al. used regularized logistic regression and boosting models to predict preterm births from medication and procedure associated data. Deep learning techniques have excelled in offering broad approaches to integrate time series and structured variables in clinical data for model structures, according to empirical evidence. Elman RNN and LSTM were proven to perform well for vital signs, although CNN and LSTM are particularly good at encoding temporal information.

Clinical diagnostics are successfully performed using deep CNNs and it was demonstrated by Brummel et al. (2017) that a hierarchical model with attention and GRU cells functions effectively with discharge notes, despite the fact that their purpose is ICD assignment of the present encounter rather than event prediction. Fiterau et al. (2017) proposed techniques to explicitly describe interaction between static information (such as age, gender, etc.) and sequential inputs for integrating mix-type inputs. Although it has been suggested that simply concatenating static covariates to sequential covariates is inefficient for learning and prone to

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/Y49NC

overfitting, static covariates are added to the input of the gates functions in LSTM and provided as parameters to the convolution function throughout time in CNN.

Clinical intervention integrating structured data and notes was anticipated by Suresh et al. (2017). Latent Dirichlet Allocation was used to convert clinical narrative notes into a 50-dimensional vector of topic proportions for each note, and static factors were duplicated across time. Thereafter, all inputs were joined together. Their empirical findings demonstrated that, given the identical input, the LSTM/CNN-based model significantly outperformed the logistic regression baseline. In their research, Tran et al. likewise employed free-text data, but they merely retrieved uni-grams from the notes after deleting stop-words. Choi et al. (2017) used a hierarchical structure from the medical term ontology via an attention method to provide auxiliary information. The goal is to address the small sample problem for some low-level codes.

Although their empirical findings appear to suggest that for some tasks, merely moving up to higher level codes or using an RNN with GloVe embeddings can produce results that are equivalent, Pyysalo and Ananiadou (2013) trained a set of word2vec embedding using PubMed data with the purpose of learning clinical text representation. In addition, Wu et al. (2017) presented StarSpace as a framework for general-purpose representation learning. The sum of the embeddings of the features that make up an entity is used to represent it. These embeddings are trained by optimising a loss function that contrasts pairs of related entities with sampling negative pairings. Users define the "label" to assess similarity; for example, a sentence and an article topic are similar pairs, whereas other themes are unfavourable pairs. The task of information retrieval ends up being the most pertinent to the goal of note representation.

3. PROPOSED METHODOLOGY

The electronic healthcare record collected from various resources such as MIMIC3 and CDSS contains many irrelevant records, which as to be cleared for future discomfort. The feature selection process followed after the pre-processing technique improves the quality of data. The procedure followed after the feature selection is classification process in analyzing stage. The visualization process remains the final part of the research work, where the output of the discussed patient future disease is shown. The detailed representation of the proposed model is given in the figure 1.

ISSN 1533-9211
DOI 10.17605/OSF.IO/Y49NC

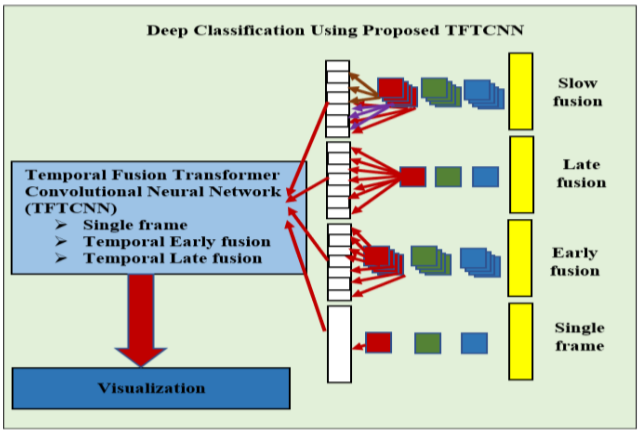

Figure 1: Proposed TFTCNN process diagram

3.1. Electronic Healthcare Records

The MIMIC3 Clinical Dataset records 61,532 ICU stays divided among 58,976 hospital admissions, themselves distributed among 46,520 subjects from Beth Israel Deaconess Medical Center, and maintained by MIT [16]. Records primarily consist of vital sign data, lab results, and the time of observation. This dataset, however, has multiple issues which need to be addressed. Events are irregularly sampled, include outlier data values, and can be entirely missing for some features and patients. Additionally, the same feature can be assigned multiple codes, which further complicates any processing.

Clinical Decision Support System CDSS Dataset are 4920 patients' history collected hospital admission records. The record set maintains 133 attributes denoting various symptoms and meditation given for the particular disease. The same record of 41 for testing purpose are divided from 4920 records of patients.

3.2. Proposed Temporal Fusion Transformer Convolution Neural Network (TFTCNN)

The identification process followed during the future disease occurrence in human body is the most important and final stage of this research work. The collected MIMIC3 and CDSS dataset from various resources are different from each other with few attributes. Earlier stage of this research work focuses on removing the irrelevant information from collected dataset as well as to improve the efficiency and quality of the collected dataset. The procedure followed after enhancing the quality of the pre-processed data is mostly avoided by many researchers, but in this research work feature extraction and selection process is carried out with proposed Conjugate Gradient and Radial Basis Function Networks (CGRBFN) algorithm.

The classification stage of the research work uses a unique way of implementing Temporal Fusion a time series Prediction procedure with the combination of Deep Learning technique. The convolutional neural network (CNN) deep learning technique implemented along with the time series fusion transformer plays a major role in reducing the classification time as well as to improve the accuracy of prediction process. The steps followed in the convolutional neural

ISSN 1533-9211

DOI 10.17605/OSF.IO/Y49NC

network-based techniques are divided into single frame, early fusion, late fusion and slow fusion with time series temporal transformer technique.

The fundamental building blocks specialize in identifying various features or trends in the time series, including a temporal multi-head attention block that distinguishes the long-term patterns the time series might include and priorities the most important patterns. Each head of attention may concentrate on a distinct temporal pattern. Long short-term memory (LSTM) blocks are used to recognize relationships of time steps with their surrounding values (whereas long-range relationships are left to the attention heads). LSTM sequence-to-sequence encoders/decoders to summaries shorter patterns. Gated residual network (GRN) blocks, are used to remove unnecessary or unneeded inputs during processing of data. In order to avoid overfitting, nodes can potentially be dropped arbitrarily. Because it combines these specialized layers to understand the interactions along the time axis, the temporal fusion decoder earns its name.

The used attributes are 32 in numbers, which is very closely associated with the future disease prediction patterns in patients. The feature selection process implemented during the procedure allows the dataset to scrutinize the collected data into more comfortable data for the prediction process. The procedure followed during the temporal fusion transformer convolutional neural network (TFTCNN) is very unique in implementing time series problem solving technique in various convolutional levels.

3.3. Temporal Fusion

EMR data differ greatly in temporal extent from images and documents, which can be considered as still, and the temporal connectedness is also crucial for the prediction. Each data sample is viewed as a collection of brief, predetermined sub-frames as in the figure 1, which is differentiated with different color. Since each sub-frame consists of a number of consecutive time periods, implementation can increase the model's connection in the temporal dimension to uncover temporal properties.

The temporal fusion can be provided with three main groups for connectivity pattern categories, which is given below. The three suggested models all rely on fusing data across temporal domains collected from different attributes, this can be done either early in the network by extending the first layer convolution filters or late by placing two distinct single-frame networks and fusing their outputs later in the processing of convolutional layers attributes.

Single-frame: The single frame temporal fusion explains about the static matrix combinations of different attribute record collected from the Electronic Healthcare Record. The procedure followed for the normalization layer is something different from other layer formations. The understandability of the layers are very clear in static representation and the accuracy of the single frame classification is better compared with other layer formations. The single frame design is mostly deployed for improving the accuracy of the overall proposed model.

Temporal Early Fusion: The Early Fusion addon instantly merges data at the fundamental event feature level from across an entire time window. The addition or combining every instant

The Seybold REPORT
ISSN 1533-9211
DOI 10.17605/OSF.IO/Y49NC

together makes the layer formation as very successful one. Each attribute is instant is associated with final frame formation and connected with the proposed convolutional method for acquiring the necessary information. This is accomplished by expanding the filters on the single-frame model's first convolution layer to cover the number of sub-frames k.

Temporal Late Fusion: The late fusion model tries to connect the entire final layer into the complete frame. The procedure follows in later fusion creates the connection for final instance to every frames in the frame sequence. The collection of single frame networks are placed together as fully connected layers and used as a single stream of message passing of information sharing. The analyzing of the layers is identifying with the single point of contact and makes the late fusion procedure as successful one. Through the layers connections happens after the layer's separations and instance identifications, it is considered as delayed fusion or late fusion. But late fusion makes it simple to identify trends in each sub-frame.

3.4. Temporal Transformer

A unique encoder-decoder model called Transformer is based on the attention mechanism and completely eliminates recurrent neural networks, which are capable of efficiently computing the sequence. The basic mechanism behind the temporal transformer is to take the combination of the layer frames into next level, which is understandable and easily accessible for convolutional network algorithms. The canonical transformer uses a stacked encoder- and decoder-layer encoder-decoder arrangement. Encoder layers are divided into two sublayers, self-attention and a position-wise feed-forward layer.

A vector is created by the encoder and fed to the decoder. Encoder-decoder attention and a position-wise feed-forward layer come after the three sublayers that make up the decoder layers, to avoid absorbing knowledge about upcoming output positions during training, the decoder uses masking in its self-attention. Transformer uses residual connections surrounding each of the sublayers, followed by layer normalization, to speed up training speed and convergence. Unlike batch normalization, which creates new dependencies during training, layer normalization calculates the normalized statistics from the total of inputs to the neurons in the hidden layer.

It has a positive impact on convolutional neural network and can increase the model's capacity for generalization. It has also been applied to models with transformers. Prior to the first layer, positional encoding based on sinusoids of various frequencies is added to the input elements of the encoder and decoder. The proposed TFTCNN that sinusoidal position encodings would aid the model's generalization to sequence lengths not encountered during training, in contrast to learnable or absolute position representations. The sine and cosine functions can be used to express the positions of the elements in a time series using the following formula:

$$\mathrm{PE}(\mathrm{pos}, 2\mathrm{i}) = \sin\left(\frac{\mathrm{pos}}{10^{8\mathrm{i}/\mathrm{d}_{\mathrm{model}}}}\right) \quad (1)$$

$$\mathrm{PE}(\mathrm{pos}, 2\mathrm{i}+1) = \cos\left(\frac{\mathrm{pos}}{10^{8\mathrm{i}/\mathrm{d}_{\mathrm{model}}}}\right) \quad (2)$$

ISSN 1533-9211

DOI 10.17605/OSF.IO/Y49NC

where i is the measurement and pos is the position. In other words, a sinusoid corresponds to each dimension of the positional encoding. Several attention heads are used by self-attention sublayers. Results from each head are combined to create the sublayer output, which is then processed using a parameterized linear transformation. Each attention head computes a new sequence z = (z1,..., zn) of the same length from an input sequence x = (x1,..., xn) of n elements. The weighted total of the linearly processed input elements is determined for each output element zi:

$$Z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V) \tag{3}$$

Each weight coefficient α_{ij} is computed using a softmax function:

$$\alpha_{ij} = \frac{e_{ij}}{\sum_{k=1}^{n} \exp e_{ik}} \tag{4}$$

In addition, eij is computed using a compatibility function that compares two input elements:

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K)^T}{\sqrt{d_z}} \tag{5}$$

where W^Q, W^K, and W^V are parameter matrices and $x_i W^Q$, $x_i W^K$, and $x_i W^V$ produce the three abstractions Q, K, and V that are useful for calculating self-attention. Simply expressed, Q stands for the query vector, K for the vector representing the correlation between the information being questioned and the context, and V for the query vector. The compatibility function, which determines how similar two items are and permits effective computing, was decided to use scaled dot product.

3.5. Visualization and comparison

Visualization and comparison are considered as final stage of this research work, were final collected weighted and information through proposed TFTCNN is shown with various confusion matrices. The true positive, True Negative, False Positive and False Negative rate calculations are given for finding out the performance of the proposed TFTCNN based model. The accuracy, precision, Recall and F-score are calculated from confusion matrices.

The same strategy used for testing the efficiency of the proposed model is tested with different existing algorithms for checking the efficiency of the existing algorithms. The comparison between proposed and existing algorithms are calculated with the measurements tabulated in results and discussion session. The following are few existing algorithms used in this research work for making comparison with proposed TFTCNN neural network model.

SVM (Support Vector Machine): The mathematical technique of creating a collection of hyper-planes in the high-dimensional space described by the predictors is known as a support vector machine. Generally speaking, greater margins could equal smaller generalization errors, hence the optimal separation can be attained if the hyper-planes have the greatest distance (functional margin) from the closest training points of any class.

KNN (Nearest Neighbor): A query point is assigned the class that has the most representatives in its geometrical hyper-space "vicinity" in the neighbors-based classification approach, which

ISSN 1533-9211
DOI 10.17605/OSF.IO/Y49NC

computes the labelling of each point in the parameters' hyper-space based on a simple majority vote of the nearest known (previously labelled) neighboring points. The most popular method is k-NN, where "k" is the quantity of nearby already-labeled sites that will participate in the "voting" process. Various applications of this technique use different choices of k, as well as different methods of calculating the geometric point-to-point distance (such as Euclidean, "Manhattan," etc.), as well as different weights for the contributions of the neighbors, such as proportional to the inverse of the distance.

NB (Naive Bayes): The use of Bayes' theorem is the foundation of naive Bayes approaches. The assumption of independence between every pair of attributes is referred to as the "naivety" of the technique. Naive Bayes classifiers have been effective in numerous real-world contexts, including document classification and spam filtering, despite their ostensibly overly simplistic approach. In comparison to other approaches, naive Bayes classifiers are very quick and require minimal training data to predict the essential parameters [7]. As a one-dimensional distribution, each distribution can be separately calculated (decoupling). This aids in addressing problems brought on by the "curse of dimensionality" (various phenomena that arise when analyzing data in high dimensional spaces).

LR (Logistic Regression): Despite its name, logistic regression is a linear model that is frequently employed for categorization. The terms "logit," "maximum-entropy classification (MaxEnt)," and "loglinear classifier" are frequently used in the literature to refer to logistic regression. The odds of the potential outcomes are modelled using a logistic function in the model.

4. RESULTS AND DISCUSSION

The Electronic Healthcare Records collected from two different sources such as MIMIC3 and CDSS consist of administrative information, Patient demographics information, Progress notes, medical histories, Vital signs, Medications, Allergies, Diagnoses, Lab test, Test results, Immunization dates and Blood Pressure. The collected dataset also has the information about the symptoms of Itching, Skin rash, Nodal skin eruptions, Continuous, sneezing, Shivering, Chills, Joint pain, Stomach pain, Acidity and Ulcers on tongue.

The proposed TFTCNN is tested with existing algorithms such as Support Vector Machine, K-Nearest Neighbor, Naive Bayes and Logistic Regression. The performance of the proposed TFTCNN shows best performance compared with other existing models table 1.

The Seybold REPORT

ISSN 1533-9211

DOI 10.17605/OSF.IO/Y49NC

Table 1: Performance matrices for proposed TFTCNN

Model	Performance			
	MIMIC3		CDSS	
	AUC	MCC	AUC	MCC
Support Vector Machine	0.610	0.336	0.400	0.225
K- Nearest Neighbor	0.684	0.470	0.473	0.260
Naive Bayes	0.607	0.330	0.406	0.120
Logistic Regression	0.622	0.355	0.411	0.144
Temporal Fusion Transformer Convolutional Neural Network (TFTCNN)	0.647	0.397	0.436	0.186

Area Under Receiver Operating Characteristic Curve (AUC): The area under a receiver operating characteristic is abbreviated as AUC ROC. The area under the ROC curve is one of the most used qualities functionals in binary classification issues. The explanation starts with the definition of new terminology using confusion matrices as follows

$$\frac{\mathrm{TPR}*\mathrm{FPR}}{2} + \mathrm{TPR} * (1 - \mathrm{FPR}) + \frac{(1-\mathrm{TPR})*(1-\mathrm{FPR})}{2} = \frac{1+\mathrm{TPR}-\mathrm{FPR}}{2} \qquad (6)$$

Matthews Correlation Coefficient (MCC): Instead, the Matthews Correlation Coefficient (MCC) is a more dependable statistical measure that only yields a high score if the prediction performed well in each of the four categories of the confusion matrix (true positives, false negatives, true negatives, and false positives), proportionally to the size of the dataset's positive and negative elements.

$$\mathrm{MCC} = \frac{\mathrm{TP}*\mathrm{TP}-\mathrm{FN}*\mathrm{FP}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FP})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \qquad (7)$$

The Support Vector Machine (SVM) The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes, allowing us to quickly classify fresh data points in the future. A hyperplane is the name given to this optimal decision boundary. SVM selects the extreme vectors and points that aid in the creation of the hyperplane. These extreme situations are called as support vectors, and consequently method is termed as Support Vector Machine. The iterations followed during SVM takes values for C = 1.0, Cache_size = 200, Class_weight = None, Coef0=0.0, degree = 3, gamma = 'auto', kernel = 'rbf', shrinking = True and Tot =0.001.

K- Nearest Neighbor (KNN) The kNN algorithm, sometimes referred to as KNN or k-NN, is a supervised learning classifier that employs proximity to produce classifications or predictions about the grouping of a single data point. Although it can be applied to classification or regression issues, it is commonly employed as a classification algorithm because it relies on the idea that comparable points can be discovered close to one another. The attributes selected for the iterations are listed as patch_size = 30, metric = 'Itching', nearest_attributes = 5 and weights = 'uniform'.

ISSN 1533-9211
DOI 10.17605/OSF.IO/Y49NC

Naive Bayes Naive Bayes is a fantastic illustration of how the most straightforward answers are frequently the most effective. Despite recent developments in machine learning, it has shown to be not only quick, accurate, and dependable but also simple. It has been used successfully for many things, but it excels at solving natural language processing (NLP) issues. The Bayes Theorem is the foundation of the probabilistic machine learning method known as Naive Bayes, which is utilized for a variety of classification problems. The iterations carried out in the implementation uses attribute selection as priors = 'none'.

The final proposed Temporal Fusion Transformer Convolutional Neural Network (TFTCNN) is time series-based technique implemented with convolutional neural network idea for deter performance. The procedure used in the algorithm creates different frames and layer formations for better classifications procedures. The usage of attribute selected for classification procedure followed during the iterations as C= 1.0, Class_weight = None, dual = False, fit_intercept = True, intercept_scaling = 1, max_iteration=100, multi_class = 'ovr', n_jobs = 1, penalty = '12', random_state = None, solver = 'liblinear', tot=0.0001, verbose = 0 and warm_start =False.

This particular part of the results discusses about qualitative discussion of the TFTCNN-based model's efficacy with other existing classification models. The result shown that the higher-order temporal event linkages can be completely utilized by the proposed framework to yield meaningful phenotypes. The differences between different approaches using each phenotype are observed. It is challenging to visualize the feature maps with deconvolutional neural networks due to model construction constraints. The technique used in the monitoring and visualization of the proposed TFTCNN clearly used for the analyzing neural activity.

Table 2: Training data variations in prediction AUC and Standard Deviation for MINIC3

Model	**60%**	**70%**	**80%**	**90%**
Support Vector Machine	0.4247± 0.111	0.5034± 0.096	0.5655± 0.075	0.6289± 0.052
K- Nearest Neighbor	0.4287± 0.106	0.5093± 0.098	0.5695± 0.075	0.6300± 0.051
Naive Bayes	0.4424±0.117	0.5242± 0.099	0.5825± 0.072	0.6579± 0.053
Logistic Regression	0.4372± 0.105	0.5319± 0.092	0.5973± 0.065	0.6685± 0.049
Temporal Fusion Transformer Convolutional Neural Network (TFTCNN)	0.4425± 0.107	0.5058± 0.088	0.5799± 0.065	0.6375± 0.052

The AUC and Standard Deviation for the dataset MIMI3 and CDSS are divided into training and testing set, were table 2 and table 3 shows the significant difference in various stages of the training set. The variations of the dataset are divided into 60%, 70%, 80% and 90% for each dataset.

ISSN 1533-9211
DOI 10.17605/OSF.IO/Y49NC

Table 3: Training data variations in prediction AUC and Standard Deviation for CDSS

Model	60%	70%	80%	90%
Support Vector Machine	0.2242± 0.096	0.3029± 0.081	0.3650± 0.060	0.4284± 0.037
K- Nearest Neighbor	0.2285± 0.091	0.3091± 0.083	0.3693± 0.060	0.4298± 0.036
Naive Bayes	0.2419±0.102	0.3237± 0.084	0.3820± 0.057	0.4574± 0.038
Logistic Regression	0.2367± 0.090	0.3314± 0.077	0.3968± 0.050	0.4680± 0.034
Temporal Fusion Transformer Convolutional Neural Network (TFTCNN)	0.2420± 0.092	0.3053± 0.073	0.3794± 0.050	0.4370± 0.037

The prediction process identified for the proposed TFTCNN classification algorithm with other existing algorithms are shown in table 2 and table 3, which clearly shows the best performance in TFTCNN. The top layer uses the output/activation of the neurons (after pooling) as features and gives the features weights. For the negative class and the positive class, respectively, the neurons whose output obtained the highest weights in the top layer. This allows to identify the places in the training set that the relevant neurons are highly activated in proposed TFTCNN.

Table 4: Measurement for proposed TFTCNN

Model	Sensitivity	Specificity	F-measures
Support Vector Machine	71.110	93.005	68.250
K- Nearest Neighbor	83.200	93.706	73.100
Naive Bayes	73.736	93.746	68.500
Logistic Regression	68.370	92.970	67.925
Temporal Fusion Transformer Convolutional Neural Network (TFTCNN)	84.826	94.165	78.865

Table 4 shows the result taken for measuring the sensitivity, specificity and F-measures for the proposed TFTCNN algorithm with other existing algorithms. The results are very clear to show the proposed TFTCNN algorithm shows the best result compared with other existing algorithms. After applying the pre-processing technique on collected dataset the information is scrutinized as in table 4

ISSN 1533-9211
DOI 10.17605/OSF.IO/Y49NC

Table 5: Accuracy for the proposed TFTCNN

Model	Accuracy	
	MIMIC3	**CDSS**
Support Vector Machine	0.802	0.722
K- Nearest Neighbor	0.890	0.789
Naive Bayes	0.885	0.796
Logistic Regression	0.879	0.845
Temporal Fusion Transformer Convolutional Neural Network (TFTCNN)	0.939	0.900

The accuracy description given for the described two different dataset are given in Table 5, which clearly shows the best performance in TFTCNN compared with other existing algorithms.

5. CONCLUSION

The most crucial record for identifying the future diseases that are likely to affect a patient is the electronic healthcare record gathered from their body. Several automated algorithms were used in the research's prediction process. The most common and efficient methods for solving problems are machine learning and big data analysis. This research work focuses on using machine learning techniques to identify the required attributes for the last stage of analysis. Following the data mining procedure's pre-processing technique is typically the feature selection or extraction step. Medical data sets gathered by MIMIC3 and CDSS are taken into account. Each record set's symptoms and essential characteristics are independently collected, and the suggested TFTCNN method is then used to predict the information. The end output is the mean and standard deviation for the notable dataset. The proposed TFTCNN algorithm is clearly analyzed for accuracy and time taken for the execution. The obtained accuracy for the two dataset used are more compared with the other existing algorithms.

Reference

1. Hersh, W. R. Adding value to the electronic health record through secondary use of data for quality assurance, research, and surveillance. Am. J. Manag. Care 13, 277–278 (2007).
2. Tatonetti, N. P., Ye, P. P., Daneshjou, R. & Altman, R. B. Data-driven prediction of drug effects and interactions. Sci. Transl. Med. 4, 125ra131 (2012).
3. Li, L. et al. Identification of type 2 diabetes subgroups through topological analysis of patient similarity. Sci. Transl. Med. 7, 311ra174 (2015).
4. Doshi-Velez, F., Ge, Y. & Kohane, I. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. Pediatrics 133, e54–63 (2014).
5. Miotto, R. & Weng, C. Case-based reasoning using electronic health records efficiently identifies eligible patients for clinical trials. J. Am. Med. Inform. Assoc. 22, E141–E150 (2015).
6. Bellazzi, R. & Zupan, B. Predictive data mining in clinical medicine: current issues and guidelines. Int. J. Med. Inform. 77, 81–97 (2008).

DOI 10.17605/OSF.IO/Y49NC

7. Jensen, P. B., Jensen, L. J. & Brunak, S. Mining electronic health records: towards better research applications and clinical care. Nat. Rev. Genet. 13, 395–405 (2012).

8. Dahlem, D., Maniloff, D. & Ratti, C. Predictability bounds of electronic health records. Sci. Rep. 5, 11865 (2015).

9. Wu, J. L., Roy, J. & Stewart, W. F. Prediction modeling using EHR data: challenges, strategies, and a comparison of machine learning approaches. Med. Care 48, S106–S113 (2010).

10. Weiskopf, N. G., Hripcsak, G., Swaminathan, S. & Weng, C. Defining and measuring completeness of electronic health records for secondary use. J. Biomed. Inform. 46, 830–836 (2013).

11. Weiskopf, N. G. & Weng, C. Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. J. Am. Med. Inform. Assoc. 20, 144–151 (2013).

12. Bengio, Y., Courville, A. & Vincent, P. Representation learning: a review and new perspectives. IEEE T. Pattern Anal. Mach. Intell. 35, 1798–1828 (2013).

13. Jordan, M. I. & Mitchell, T. M. Machine learning: trends, perspectives, and prospects. Science 349, 255–260 (2015).

14. Huang, S. H. et al. Toward personalizing treatment for depression: predicting diagnosis and severity. J. Am. Med. Inform. Assoc. 21,1069–1075 (2014).

15. Lyalina, S. et al. Identifying phenotypic signatures of neuropsychiatric disorders from electronic medical records. J. Am. Med. Inform. Assoc. 20, e297–305 (2013).

16. Wang, X., Sontag, D. & Wang, F. Unsupervised learning of disease progression models. ACM SIGKDD, 85–94 (2014).

17. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015).

18. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P. A. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. 11, 3371–3408 (2010).

19. Shah, N. H. et al. Comparison of concept recognizers for building the Open Biomedical Annotator. BMC Bioinformatics 10, S14 (2009).

20. Musen, M. A. et al. The National Center for Biomedical Ontology. J. Am. Med. Inform. Assoc. 19, 190–195 (2012).

21. Jonquet, C., Shah, N. H. & Musen, M. A. The Open Biomedical Annotator. Summit on Translat. Bioinforma. 2009, 56–60 (2009).

22. Lependu, P., Iyer, S. V., Fairon, C. & Shah, N. H. Annotation analysis for testing drug safety signals using unstructured clinical notes. J. Biomed. Semantics 3, S5 (2012).

23. Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F. & Buchanan, B. G. A simple algorithm for identifying negated findings and diseases in discharge summaries. J. Biomed. Inform. 34, 301–310 (2001).

24. Cohen, R., Elhadad, M. & Elhadad, N. Redundancy in electronic health record corpora: analysis, impact on text mining performance and mitigation strategies. BMC Bioinformatics 14, 10 (2013).

25. Blei, D. M. Probabilistic topic models. Commun. ACM 55, 77–84 (2012).

26. Arnold, C. W., El-Saden, S. M., Bui, A. A. & Taira, R. Clinical case-based retrieval using latent topic analysis. AMIA Annu. Symp. Proc., 26–30 (2010).

27. Perotte, A., Bartlett, N., Elhadad, N. & Wood, F. Hierarchically supervised latent dirichlet allocation. NIPS, 2609–2617 (2011).

ISSN 1533-9211

DOI 10.17605/OSF.IO/Y49NC

28. Bisgin, H., Liu, Z., Fang, H., Xu, X. & Tong, W. Mining FDA drug labels using an unsupervised learning technique - topic modeling. BMC Bioinformatics 12, S11 (2011).

29. Blei, D. M., Ng, A. Y. & Jordan, M. I. Latent Dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003).

30. Cowen, M. E. et al. Casemix adjustment of managed care claims data using the clinical classification for health policy research method. Med. Care 36, 1108–1113 (1998).

31. Larochelle, H., Bengio, Y., Louradour, J. & Lamblin, P. Exploring strategies for training deep neural networks. J. Mach. Learn. Res.10, 1–40 (2009).

32. Breiman, L. Random forests. Mach. Learn. 45, 5–32 (2001).

33. Fernandez-Delgado, M., Cernadas, E., Barro, S. & Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 15, 3133–3181 (2014).

34. Manning, C. D., Raghavan, P. & Schütze, H. Introduction to Information Retrieval. (Cambridge University Press, 2008).

35. Helmstaedter, M. et al. Connectomic reconstruction of the inner plexiform layer in the mouse retina. Nature 500, 168–174 (2013).

36. Ma, J. S., Sheridan, R. P., Liaw, A., Dahl, G. E. & Svetnik, V. Deep neural nets as a method for quantitative structure-activity relationships. J. Chem. Inf. Model 55, 263–274 (2015).

37. Leung, M. K. K., Xiong, H. Y., Lee, L. J. & Frey, B. J. Deep learning of the tissue-regulated splicing code. Bioinformatics 30, 121–129 (2014).

38. Xiong, H. Y. et al. The human splicing code reveals new insights into the genetic determinants of disease. Science 347, 144–151 (2015).

39. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nature Biotech. 33, 831–838 (2015).

40. Liang, Z., Zhang, G., Huang, J. X. & Hu, Q. V. Deep learning for healthcare decision making with EMRs. IEEE BIBM, 556–559 (2014).

41. Hinton, G. E. & Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. Science 313, 504–507 (2006).

42. Lasko, T. A., Denny, J. C. & Levy, M. A. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. PLoS One 8, e66341 (2013).

43. Kennedy, E. H., Wiitala, W. L., Hayward, R. A. & Sussman, J. B. Improved cardiovascular risk prediction using non-parametric regression and electronic health record data. Med. Care 51, 251–258 (2013).

44. Hui, L., Xiaoyi, L., Ramanathan, M. & Aidong, Z. Prediction and informative risk factor selection of bone diseases. IEEE/ACM T. Comput. Biol. Bioinform. 12, 79–91 (2015).

45. Perotte, A., Ranganath, R., Hirsch, J. S., Blei, D. & Elhadad, N. Risk prediction for chronic kidney disease progression using heterogeneous electronic health record data and time series analysis. J. Am. Med. Inform. Assoc. 22, 872–880 (2015).

46. Perotte, A. et al. Diagnosis code assignment: Models and evaluation metrics. J. Am. Med. Inform. Assoc. 21, 231–237 (2014).

47. Gottlieb, A., Stein, G. Y., Ruppin, E., Altman, R. B. & Sharan, R. A method for inferring medical diagnoses from patient similarities. BMC Med. 11, 194–203 (2013).

48. Yao, L. X., Zhang, Y. Y., Li, Y., Sanseau, P. & Agarwal, P. Electronic health records: Implications for drug discovery. Drug Discov. Today 16, 594–599 (2011).