2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) | 979-8-3503-1199-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/IPDPSW59300.2023.00036

Using Hyperdimensional Computing to Extract Features for the Detection of Type 2 Diabetes

Neftali Watkinson
Department of Computer Science and Engineering
University of California, Riverside
Riverside, USA
neftaliw@ucr.edu

Divya Devineni
UCI Medical Center
University of California, Irvine
Irvine, USA
devineni@hs.uci.edu

Victor Joe
UCI Medical Center
University of California, Irvine
Irvine, USA
vcjoe@hs.uci.edu

Tony Givargis
Department of Computer Science
University of California, Irvine
Irvine, USA
givargis@uci.edu

Alexandru Nicolau
Department of Computer Science
University of California, Irvine
Irvine, USA
nicolau@ics.uci.edu

Alexander Veidenbaum
Department of Computer Science
University of California, Irvine
Irvine, USA
alexv@ics.uci.edu

***Abstract*—Diabetes impacts around 8% of the world's population, with Type 2 diabetes comprising up to 90% of cases. This chronic disease is characterized by a metabolic resistance to insulin which results in a high blood sugar level and increased potential for serious health complications. Preventative medicine and the detection of genetic predisposition play a key part in successful treatment. Although several factors have been identified as possible indicators of underlying diabetes, they are not the same in every patient. There have been different approaches to producing predictive models that could help identify risk of onset diabetes. Models built using Machine Learning algorithms have showed promise in the past in detecting relevant features in sample datasets with data from patients at risk of developing diabetes. However, overall performance has not been consistent across datasets. In this paper we describe a feature extraction approach using Hyperdimensional Computing as a tool for improving already existing classification models. We tested our approach using two public datasets and compare across several state of the art models. Our approach improves poor performing models while fine tuning models with a high classification accuracy.**

***Index Terms*—diabetes, hyperdimensional computing, deep learning, classification**

I. Introduction

Diabetes mellitus (commonly known as diabetes), is a metabolic disorder that affects a considerable percentage of the world's population, with some reports suggesting that over 8% of the world's population is living with diabetes [1]. Type 2 diabetes, the most common form (up to 90% of all cases of diabetes), is characterized by a metabolic resistance to insulin, a natural hormone produced by the pancreas, and relative insulin deficiency. This results in a high blood sugar level and health complications that include heart disease, blindness, poor circulation, and death [2]. In most cases, type 2 diabetes is treatable and a number of lifestyle-related approaches can be effective in preventing diabetes or delaying disease progression. Sedentarism, high body fat, and poor nutrition rich in sugars and carbohydrates are heavily linked to an increased risk of developing type 2 diabetes. However, genetic predisposition can be a determining factor that is difficult to detect. While individual cases might not be linked to a broader predisposition, many studies have identified key populations with a high tendency for type 2 diabetes. One of the study populations is that of the Akimel O'odham or commonly known as Pima [3].

In the early 1970's, a large study of the Pima Population sponsored by the National Institute of Diabetes and Digestive and Kidney Diseases was conducted to identify the main factors behind the high index of diabetes registered in this group. While obesity and genetics were identified as key indicators, other factors were identified as predictors too. The resulting dataset from this study has been the focus of medical and machine learning (ML) research for many years. Smith, et al. [4] developed an adaptive neural network called ADAP that was applied to a subset within the Pima population dataset. These analyses focused on predicting future diabetes in women over the age of 21 among those who were not diagnosed with diabetes within the next year. The goal was to develop a prediction algorithm to identify women who would be diagnosed with diabetes within one to five years after the initial assessment.

A newer dataset was presented by Islam, et al. [5]. This dataset was constructed by surveying patients at the Hospital of Sylhet, Bangladesh, who had symptoms that were considered indicative of diabetes. The outcome was validated by the result of medical assessment. The authors of the dataset use a supervised ML algorithm to classify patients as being diabetic or not at the time of medical assessment. Unlike the Pima dataset, the outcome of the Sylhet dataset was immediate (diabetes testing performed at the same visit when the features are collected).

Both datasets show that there's a correlation between vital signs, lab results and genetic predisposition, to a positive diagnosis of diabetes. Since then, other ML algorithms have

979-8-3503-1199-0/23/$31.00 ©2023 IEEE
DOI 10.1109/IPDPSW59300.2023.00036

been used with different levels of success at detecting such correlation. The purpose of this work is to use a novel approach based on Hyperdimensional Computing (HDC) to improve predictive models without adding a large computational overhead. HDC is a neuroinspired architectural approach to pattern recognition [6]. The full scope of HDC proposes replacing the Von-Neumann models that focus on translating problems into models that can be executed using binary computing, for a new model of computation that relies on sparse distributed memory. Even on current architectures, HDC has shown great potential for classification problems [7].

A. Related work

On the intersection of diabetes and ML, most research is focused on predicting diabetes before the patient is diagnosed with the condition. With the greater availability of data and the accuracy that deep learning achieves, some of the research has seen real life implementation [8]–[10]. In the specific case of the Pima dataset, most recent research achieves up to 98% validation accuracy using deep learning [11]. It is important to note, however, that due to the size of the dataset, validation accuracy is unlikely to translate into testing accuracy. For example, one of the models we developed that used a Sequential Neural Network without early stopping, achieved 100% validation accuracy, but very poor testing accuracy. This is most likely due to overfitting. Therefore, it is difficult to cross compare work done for Pima since they vary on how they handle the data. However, models closely comparative to ours that do little preprocessing of data have a testing accuracy range between 67% and 85% [12]–[16].

Park et al. [17] used a sequential neural network, similar to the one we used in this work, but applied to a different dataset. They achieved an overall classification accuracy of 86% in detecting diabetes. There's promising research with bigger datasets that is being translated into real world applications [18]. The most resilient ones are self-improving and self-sustainable by feeding from the data they process. This is what allows novel approaches achieve the development study phase of research and eventually implementation.

HDC has had good results within bioinformatics. The work of Rahimi, et al. [19], [20] uses data from non-invasive electrode's to model brain activity and predict the subject's intentions. They achieve a 5% improvement in accuracy over ML approaches. Similarly, Imani, et al. [21] uses HD computing for DNA modeling, achieving over 99% accuracy.

In this work, we use HDC on the features present in each dataset to encode each instance (patient subject) as a multi-dimensional vector (hypervector) to predict the future onset of diabetes (for the Pima dataset) or to detect the presence of diabetes (for the Sylhet dataset). We describe a purely HDC model that uses Hamming distance [22] as the classification metric and a hybrid HDC with Deep Learning (HDC+DNN) model that uses hypervectors as input for ML models. The HDC model achieves up to 79% classification accuracy on the Pima dataset and 96% classification accuracy on the Sylhet dataset. The HDC+DNN model achieves 89% validation and testing accuracy on the Pima dataset and 97% validation and testing accuracy on the Sylhet dataset. Finally, we compared our models with state of the art models, using both regular features and hypervectors. We discuss that adding hypervectors to existing models has the potential of improving their validation accuracy up to 10%.

II. Methodology

For this work, we first encoded the features into binary hypervectors with ten thousand (10k) bits. The full and detailed explanation for the relationship between dimensionality and representation can be found in Kanerva's work [6] but here is a brief and paraphrased explanation of it. With 10k bits we have $2^{10,000}$ different hypervectors that we can generate to represent unique data points by flipping digits. Our data points can be represented in this high dimensional *hyperspace* where we can exploit certain properties, especially those related to clustering and proximity of relevant hypervectors. For example, for any given point, half of the space is closer than 0.5 (that is, 5,000 bits or less are different) and the other half is farther away. However, at a distance of 0.47 only a thousand-millionth is closer. This distribution makes the model robust and tolerant to noise since it is easier to separate correlated hypervectors from those that are distant in the hyperspace. While dimensions of 20,000 or 30,000 share similar properties, through informal experiments, we didn't see much improvement by using larger vectors.

We use binary hypervectors because binary operations on a Von Neumann architecture are easy and highly efficient. Many operations such as multiplication and addition can be applied using logical operators (such as AND, OR and XOR) and can exploit low-level parallelism. However, ternary (with values of -1, 0 and 1) and integer hypervectors could also be used [6].

After encoding our data points, we classified the hypervectors using a distance metric and later a supervised ML algorithm. The original HDC classification model relies on hamming distance to identify the output class. While euclidean distance could also be used, computing hamming distances on binary vectors is more straightforward (the distance is given by the number of bits that are different between two hypervectors). In addition to this distance based model, we used the hypervectors to train classification models based on Random Forest [23], Decision Trees [24], K Nearest Neighbors (KNN) [25], eXtreme Gradient Boosting (XGBoost) [26], CatBoost [27], Stochastic Gradient Descent (SGD) [28], Support Vector Classifier (SVC) [29], Light Gradient Boosting Machine (LGBM) [30], Logistic Regression [31], and a Sequential Deep Neural Network (Sequential NN) with a Rectified Linear Unit (ReLU) activation [32]. All of these models were used as implemented in the Scikit-Learn ML library [33].

A. Datasets

For this work we use the Pima dataset and the Syhlet dataset. We chose these two because of the work that has been done around them and because they offer different perspectives

Feature	Positive	Negative
Age	36 (21-60)	28 (21-81)
Pregnancies	4 (0-17)	3 (0-13)
Glucose	145 (78-198)	111 (56-197)
BMI	36 (23-67)	32 (18-57)
Skin Thickness	33 (7-63)	27 (7-60)
Insulin	207 (14-846)	130 (15-744)
DPF	0.6 (0.12-2.42)	0.47 (0.08-2.39)
Blood Pressure	74 (30-110)	69 (24-106)

TABLE I
FEATURE DISTRIBUTION FOR THE 8 FEATURES CAPTURED IN THE DATA SET. THE VALUE REPRESENTS THE AVERAGE AND INSIDE THE PARENTHESES, THE RANGE.

to the problem of predicting onset diabetes. However, it is important to understand the pecularities of both datasets.

1) Pima Dataset: The data captured by Knowler, et al. [3] included all adult members of the Pima community in Arizona. For a span of 5 years, they captured relevant data including family history, plasma glucose concentration, and physical measurements such as body mass index. The derived dataset, made initially available by Smith, et al. [4], considered additional qualifiers:

- The subject is female.
- The subject is over 21 years old.
- Using a glucose tolerance test (GTT), diabetes was either detected within the next five years (positive) or GTT didn't detect any in five years or more.
- If diabetes occurred within one year or less of the sampling date, then the data for that subject was discarded. This is done as a curating measure to reduce the number of patients who were misdiagnosed as non-diabetic and the time of collection.

The dataset contains several entries with missing data. To deal with this drawback we removed subjects that had missing data, ending up with 262 patients in the negative class and 130 in the positive class. Table I describes the value distribution per feature. These were selected due to their relevance for diagnosing type 2 diabetes. Age and Body Mass Index (BMI) have been widely documented as correlated to type 2 diabetes [34]. Glucose and Insulin concentrations are obtained through the Plasma Glucose Concentration at 2 Hours in an oral Glucose Tolerance Test (GTT) which is a trusted source for diagnosis. Number and quality of pregnancies, as well as blood pressure (diastolic) are indicators of diabetes as well [35]. Diabetes degree function and Skin Thickness were novel observations at the time of the study:

- Skin thickness: Triceps Skin Fold Thickness has been used as an assessment of proper nutrition since the mid-1970s [36]. The measurement is done around the upper back of the left arm. This value is often used along with BMI for assessing a patient's fat concentration. Since type 2 diabetes is heavily linked with obesity, skin thickness is as accurate as using other bodily measurements.
- Diabetes Pedigree Function: DPF was developed by Smith, et al. [4] to quantify the family history with type 2 diabetes. For each subject, DPF is computed as:

$$DPF = \frac{\sum_i (K_i(88 - ADM_i) + 20)}{\sum_j (K_j(ALC_j - 14) + 50)}$$

Where *i* measures the ranges of all relatives who had developed diabetes before the examination date, *j* for all the relatives who didn't develop diabetes. K is the percentage of genes shared by the relative (0.5 for a parent or sibling, 0.25 for half sibling, grandparent or a parent's sibling, and 0.125 for cousins and parent's half siblings). Relative's age of diabetes mellitus (ADM) is the age of a relative with diabetes and Relative's age of cleared diagnosis (ACL) for a non-diabetic relative. Constants 88 and 14 are for normalizing the function according to the maximum and minimum relative ages. The constants 20 and 50 adjust the function to emphasize old relatives without diabetes and young relatives with diabetes.

The effectiveness of these two features in predicting diabetes has been documented since then [37].

In order to deal with the issue of missing data, the dataset with all missing values removed is called Pima R. We did a separate set of experiments using the version of the dataset generated by Artem [38] where each missing value was replaced with the median value of it's corresponding class. We call this Pima M.

2) Syhlet dataset: The Syhlet dataset was collected through questionnaires from patients of the Sylhet Diabetes Hospital in Sylhet, Bangladesh [5]. It contains 520 entries (320 positive, 200 negative). The purpose of their work was to compare the accuracy of 4 ML models (Naive Bayes, Logistic Function, Decision Tree and Random Forest) in identifying diabetes in patients that had symptoms correlated to it. unlike the Pima dataset, patients in the Sylhet dataset who are positive have already developed diabetes at the time of data extraction. Their best performing model is built using Random Forest with a 97.4% accuracy in a 10 fold cross-validation test. The features of the dataset are: age, sex, polyuria, polydipsia, sudden weight loss, weakness, polyphagia, vaginal thrush, itching, irritability, delayed healing, partial paresis, muscle stiffness, alopecia, and obesity. Other than age, all the other features are binary (sex uses 1 for Male and 2 for Female).

B. Data encoding

The features in the Pima dataset are linear. That is, the different values are in proximity to each other relative to the difference in magnitudes. Because of this, we used linear encoding for generating the hypervectors. In general, this type of encoding is used when the relative value of a feature contains crucial information about the instance [19]. For example, when analyzing age we know by intuition that a value of 45 is closer to 50 than it is to 70. The linear encoding algorithm for binary 10k long hypervectors is as follows:

1) For each feature, identify *min(V)* and *max(V)* which are respectively the lowest and highest feature values.

2) Generate a random 10k binary hypervector that is partially dense (has an equal amount of 1s and 0s). This is our seed vector and is used to represent every value equal or lesser than *min(V)*. (A lesser value could be found in new data that hasn't been seen by the encode)
3) For all the remaining values, flip an equal x number of 0 and 1 bits from the seed vector according to the following formula:

$$x = \frac{k(t - \min(V))}{2(\max(V) - \min(V)))}$$

Where k is the dimensionality of the vectors, t is the target value, and V is the list of all the continuous values for a specific feature. The range is doubled so that the highest value gets a hypervector orthogonal (half of the elements are different) to the hypervector representing the lowest value.

Notice that our encoding process will generate two hypervectors that are orthogonal to each other: one for the lowest value and one for the highest value. This means that for each feature, each hypervector is at most 5,000 bits away from all other hypervectors and the hypervectors that represent neighboring values are closer to each other than to those of distant values in a linear sequence. Each feature has a different seed hypervector. Randomness is important during the encoding process, we don't want to bias the encoding towards the relevance of a subset of features.

For the Syhlet dataset, we used linear encoding for the age feature. The remaining features are binary (values can be yes or no), therefore, we generated a seed hypervector to represent 0 and an orthogonal hypervector to represent 1. The orthogonal hypervector is generated by flipping an equal number of 1's and 0's chosen randomly.

For each dataset, the hypervectors that represent the features are combined into a patient (or instance) hypervector. The encoding for each patient works as follows:

1) Compile all the feature vectors for the patient.
2) Combine all the feature hypervectors using the majority rule (also known as majority voting) on each bit.
3) Use the newly generated 10k bit hypervector to represent the patient in the hyperspace.

Majority voting works as follows:

1) For each bit of the hypervectors. Count the number of 1's and 0's.
2) Set the bit in the combined hypervector to the most common number (1 or 0) in the feature hypervectors.
3) For tie braking (when the number of 1s and 0s is equal) we chose 1 as the resulting number [39].

For example, given three feature hypervectors: *A, B* and *C*. If the first bit of A (A_0) and the first bit of B (B_0) are 1, and the first bit of C (C_0) is 0, then the first bit of the hypervector resulting from combining the three vectors will have the value of 1. An alternate approach to counting each bit is to add the respective bits, divide by the number of feature hypervectors, and round the result to 1 or 0.

C. Classification with Hamming distance

After all the patients have been encoded, we can use a distance metric to determine similarity among them. Hamming distance was originally proposed as the go to metric for binary hypervectors due to its computational efficiency [6], [39]. We used Hamming distance for classification and validate our model using leave-one-out cross-validation. The process for this validation is as follows:

1) For each patient hypervector, measure its Hamming distance to all other patient hypervectors. Record the predicted class as the known class of the closest hypervector.
2) Compare the actual class of the patient hypervector to the predicted class. If the classes are equal it is recorded as a true positive (both classes are 1) or true negative (both classes are 0). If they are not equal it is recorded as a false positive or false negative.
3) Repeat for all other patient hypervectors in the dataset.

Leave-one-out is generally the most accurate cross-validation method but also the most cost-prohibitive [40]. However, when compared to traditional ML, HDC has the algorithmic advantage that once the hypervectors are constructed there's no model that needs to be built, we only need to measure distances. Considering the relative small size of our datasets, leave-one-out is feasible and cost-effective.

D. Classification with a Deep Neural Network

Using Hamming distance for HDC classification has yielded classification performance metrics that matches or beats other ML approaches in other domains [21], [39], [41]–[44]. However, using hybrid approach, where the distance metric is replaced by an ML model, has the potential of yielding improved results. For this purpose we built two sequential neural networks (Sequential NN) [45], one with the original features as input (8 for Pima and 16 for Syhlet) and the other with hypervectors as input (10k). Both networks consist of two dense layers with 32 nodes and a ReLu activation function and binary output layer with a sigmoid activation function.

For validation purposes, the data is split 70/15/15 (70% is used for training, 15% for validation and 15% for testing). We ran each network for 1000 epochs with an early stopping condition (if the loss function doesn't improve across 20 consecutive epochs, the training stops). We repeated the experiment 10 times and reported on the average testing accuracy (Table II)

III. Results

A. Classification with various models

The Table II shows the testing accuracy of the Hamming and Sequential Neural Network models. The latter comparing the model built with the original features to the one that used hypervectors. For both versions of the Pima dataset, the Sequential NN model saw a considerable improvement when using hypervectors. However, with the Syhlet dataset there was no improvement. One reason for this could be

Model	Pima R		Pima M		Syhlet	
	Features	Hypervectors	Features	Hypervectors	Features	Hypervectors
Hamming	-	70.7%	-	78.8%	-	95.9%
Sequential NN	71.2%	79.6%	75.9%	88.8%	97.4%	97.4%

TABLE II
TESTING ACCURACY FOR THE HD COMPUTING BASED MODEL USING HAMMING DISTANCE AND THE SEQUENTIAL NEURAL NETWORK TRAINED ON RAW FEATURES AND ON HYPERVECTORS.

due to the added dimensionality of Syhlet's features and that Syhlet is a more balanced dataset than Pima. The Hamming distance model, as most distance-based models, will suffer if there aren't enough examples of each class to populate the space. We believe that this is why both models did better with Syhlet. The work of Artem [38] provides a comparison across popular ML models using the Pima dataset and with a focus on accuracy. We created our own models using the same ML algorithms implemented by Artem and other works [46], [47]. We evaluated the impact of adding hypervectors to these models and used them to classify each of the three datasets: Pima with missing values removed (Pima R), Pima with median values replacing the missing values (Pima M), and the Syhlet dataset. In order to normalize the results, we used the same hyper-tuning variables used in the mentioned references and modified the input for using hypervectors. Following the original validation methodology, we used 10-fold cross validation. For the ML models we ran two experiments per dataset, one with the original feature values and the second using hypervectors. Before looking at the testing performance metrics we analyzed how the training accuracy was impacted by the addition of hypervectors. Table III shows the training accuracy for each experiment.

Combining hypervectors with other models sometimes translates into a dramatic improvement (over 10% higher accuracy in the case of SGD) or reduces the accuracy of the model by 4% in the worse case. On average, hypervectors improved the training accuracy of models by 1.3%. We didn't fine-tune any of the ML models to adapt for hypervectors, therefore there is potential for further improvement.

We looked closer at the models for the Syhlet and the Pima M datasets. Table IV has the **testing** performance metrics for the Pima M datasets. We used a sample of 10% of the dataset for testing, training on the other 90%. While we still observed inconsistencies in how much improvement was obtained from using hypervectors, the combination of hypervectors with Random Forest and SVC produced the strongest performing models.

The metrics for the Syhlet dataset using the same approach are in Table V. Random Forest with hypervectors once again outperformed every other model. We include the Hamming model for reference, however the metrics for it are from leave-one-out validation. Hamming distance alone results in a relatively high accuracy. This was surprising to us since this is the most cost-effective approach among the models presented in this work.

Regarding running time, we observed that the performance of the Sequential Neural Network was similar (10 msec per epoch) using the original feature values or the hypervectors as input. On the other hand, LGBM, XGBoost and CatBoost see a major increase in computing time when using hypervectors (over 10x). We didn't observe a significant performance difference for the remaining models. We do not account for the time it takes to build the hypervectors. Maximizing performance of HDC is beyond the scope of this work.

It seems that the amount of data and number of available features has an influence on whether hypervectors will improve a classification model or not. With lesser amounts of data, deep learning strategies struggle to perform well, a phenomenon that has been seen in other data-constraint problems [48]. But when more data is available, as is the case with the Syhlet, the added dimensionality from the hypervectors doesn't contribute much towards improved accuracy. We cannot say the same about the Pima M datset since the synthetic features could be adding bias to the data. Further evaluation is needed.

In the end, the decision of using hypervectors would depend on the application's constraints such as the amount of data available, the computing requirements, and the ML model being used. The performance of Random Forest was surprising. An explanation for its performance could be due to Random Forest being an ensemble algorithm. The bagging step that happens within the model might benefit by the added dimensionality of the hypervectors.

B. Significance for Medicine

Clinicians diagnose type 2 diabetes through a combination of two abnormal laboratory results and symptoms of high blood sugar. Hemoglobin A1C (HbA1c) is a test that measures blood sugar levels on average from the past 3 months through determining the amount of red blood cells that have sugar attached to their hemoglobin. Clinicians diagnose diabetes at a HbA1c level of greater than or equal to 6.5%. Fasting plasma glucose levels of greater than or equal to 126 mg/dl or random plasma glucose levels of greater than or equal to 200 mg/dl are also diagnostic of diabetes. Lastly, an oral glucose tolerance test (OGTT), a test that shows how the body processes sugar, of greater than or equal to 200 mg/dl is indicative of diabetes [49].

Clinicians can also predict onset of diabetes using these tools. HbA1c between 5.7% and 6.4%, fasting plasma glucose levels between 100 mg/dl and 125 mg/dl, or OGTT of 149 mg/dl and 199 mg/dl are all indicative of prediabetes. However, health care providers examine risk factors and genetic predispositions in warning their patients of developing diabetes. A variety of risk factors contribute to type 2 diabetes including being 45 years or older, having a BMI of or equal

Model	Pima R		Pima M		Syhlet	
	Features	Hypervectors	Features	Hypervectors	Features	Hypervectors
Random Forest	78.4%	78.5%	92.0%	88.6%	98.0%	97.8%
KNN	75.9%	78.1%	91.4%	85.8%	92.9%	95.6%
Decision Tree	77.4%	76.6%	87.7%	84.5%	97.5%	96.7%
XGBoost	78.8%	77.0%	91.6%	88.8%	96.9%	97.8%
CatBoost	78.4%	77.4%	92.6%	88.8%	98.3%	97.5%
SGD	67.1%	77.7%	74.4%	87.7%	90.9%	96.7%
Logistic Regression	78.5%	77.0%	78.3%	87.5%	93.1%	96.4%
SVC	77.4%	78.1%	86.2%	87.7%	92.9%	97.5%
LGBM	78.1%	76.3%	91.1%	88.8%	96.1%	96.4%

TABLE III

TRAINING ACCURACY FOR EACH MACHINE LEARNING MODEL WITH ORIGINAL FEATURE VALUES AND WITH HYPERVECTORS

	Precission		Recall		Specificity		F1 Score		Testing Accuracy	
Model	Features	HD	Features	HD	Features	HD	Features	HD	Features	HD
Random Forest	0.829	**0.866**	0.872	**0.888**	0.650	**0.711**	0.850	**0.877**	79.66%	**83.05%**
KNN	0.793	**0.817**	**0.855**	0.827	**0.595**	0.595	**0.823**	0.822	**76.27%**	75.42%
Decision Tree	**0.817**	0.768	**0.870**	0.840	**0.634**	0.558	**0.843**	0.803	**78.81%**	73.73%
XGBoost	0.829	0.829	**0.895**	0.883	**0.667**	0.659	**0.861**	0.855	**81.36%**	80.51%
CatBoost	**0.805**	0.793	**0.868**	0.855	**0.619**	0.595	**0.835**	0.823	**77.97%**	76.27%
SGD	0.561	**0.695**	0.868	**0.934**	0.446	**0.561**	0.681	**0.797**	63.56%	**75.42%**
Logistic Regression	**0.866**	0.817	**0.877**	0.827	0.703	0.595	**0.871**	0.822	**82.20%**	75.42%
SVC	0.854	**0.866**	0.886	**0.888**	0.692	**0.711**	0.870	**0.877**	82.20%	**83.05%**
LGBM	0.817	**0.841**	**0.870**	0.863	0.634	**0.658**	0.843	**0.852**	78.81%	**79.66%**

TABLE IV

PERFORMANCE OF MACHINE LEARNING MODELS USING THE PIMA M DATASET. THE NUMBERS IN BOLD ARE THE VERSIONS OF THE MODEL (FEATURE BASED OR HD COMPUTING BASED) THAT HAD THE BEST PERFORMANCE IN THE RESPECTIVE METRIC.

	Precission		Recall		Specificity		F1 Score		Testing Accuracy	
Model	Features	HD	Features	HD	Features	HD	Features	HD	Features	HD
Random Forest	0.957	0.957	0.967	**0.989**	0.938	0.938	0.962	**0.973**	95.51%	**96.79%**
KNN	0.943	**0.956**	0.901	**0.956**	0.923	**0.938**	0.921	**0.956**	91.03%	**94.87%**
Decision Tree	**0.947**	0.946	**0.978**	0.956	0.923	0.923	**0.962**	0.951	**95.51%**	94.23%
XGBoost	**0.947**	0.918	**0.989**	0.978	**0.923**	0.877	**0.968**	0.947	**96.15%**	93.59%
CatBoost	**0.957**	0.947	0.967	**0.978**	**0.938**	0.923	0.962	**0.962**	95.51%	95.51%
SGD	**0.954**	0.880	0.729	**0.967**	**0.958**	0.815	0.827	**0.921**	83.33%	**90.38%**
Logistic Regression	0.869	**0.936**	0.945	**0.967**	0.800	**0.908**	0.905	**0.951**	88.46%	**94.23%**
SVC	0.914	**0.938**	0.934	**0.989**	0.877	**0.908**	0.924	**0.963**	91.03%	**95.51%**
LGBM	**0.938**	0.936	**0.989**	0.967	0.908	0.908	**0.963**	0.951	95.51%	**94.23%**
Hamming	-	0.984	-	0.950	-	0.975	-	0.967	-	95.96%

TABLE V

PERFORMANCE OF MACHINE LEARNING MODELS USING THE SYHLET DATASET. THE NUMBERS IN BOLD ARE THE VERSIONS OF THE MODEL (FEATURE BASED OR HD COMPUTING BASED) THAT HAD THE BEST PERFORMANCE IN THE RESPECTIVE METRIC.

to 25 kg/m2, a waist circumference of greater than 40 inches in males and greater than 35 inches in females, and lack of physical activity [3]. Other factors examined are high risk ethnic populations, family history of diabetes, history of gestational diabetes, hypertension, atherosclerotic cardiovascular disease, use of certain antipsychotics and glucocorticoids, and conditions associated with insulin resistance [50]. Healthcare providers can advise patients to modify their lifestyles and manage the severity of conditions to prevent the development of type 2 diabetes.

A clear application of our model can be in informing clinicians of the presence of diabetes in their patients, ultimately aiding in diagnosing diabetes. We can warn doctors of high-risk diabetic patients. The data can be directly fed from electronic health records, and then present a score to inform clinicians so they can manage the trajectory of their patients. This predictive score could guide preventative efforts of controlling incidence of diabetes such as weight loss, physical activity, and pharmacologic intervention for blood pressure or lipid management [51].

A flexible model, such as ours that uses HDC, can be incorporated into regular follow up visits. The model can help assess if the risk of developing diabetes has increased, decreased, or remained unchanged and inform doctors on how effective their management or intervention was in their patients. In order to explore this further we will need to collect data form individual patients and observe. Since HDC performs well with small datasets we believe that it can adapt well to tailored predictions.

IV. CONCLUSION

In this work we described a Hyperdimensional Computing approach to feature extraction that has the potential of improving ML models towards the prediction of type 2 diabetes. Our experiments included two datasets, the Pima and the Syhlet dataset. We evaluated our approach using 10 different ML

models and observed that using hypervectors instead of regular features sometimes improves classification accuracy, in some cases without adding significant computational overhead. We observed that when data is scarce, our approach has the largest positive impact in classification accuracy. We showed that a computationally efficient approach using Hamming distance was able to achieve accuracy that rivaled iterative approaches (95.9% for the Syhlet dataset compared to 97.8% from Deep Learning approaches). We also show that Random Forest using hypervectors as input produced the strongest classification model based on performance metrics during testing. For easy comparison, we used the same hyper-parameters reported in other papers that use these datasets. Further tuning and exploration needs to be done to fully understand the potential of this approach.

Future work will be centered on exploring real-life applications of an HDC approach to predicting diabetes. We believe that an HDC-based model could help in the early and in-situ detection of diabetes that could aid efforts where big data processing is not feasible nor appropriate. However, the Pima and Syhlet datasets are not representative of the data that is available in modern electronic health records. A proper study is required through appropriate data access.

REFERENCES

[1] E. R. F. Collaboration *et al.*, "Diabetes mellitus, fasting blood glucose concentration, and risk of vascular disease: a collaborative meta-analysis of 102 prospective studies," *The Lancet*, vol. 375, no. 9733, pp. 2215–2222, 2010.

[2] D. Mellitus, "Diagnosis and classification of diabetes mellitus," *Diabetes care*, vol. 28, no. S37, pp. S5–S10, 2005.

[3] W. C. Knowler, D. J. Pettitt, P. J. Savage, and P. H. Bennett, "Diabetes incidence in pima indians: contributions of obesity and parental diabetes," *American journal of epidemiology*, vol. 113, no. 2, pp. 144–156, 1981.

[4] J. W. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*. American Medical Informatics Association, 1988, p. 261.

[5] M. F. Islam, R. Ferdousi, S. Rahman, and H. Y. Bushra, "Likelihood prediction of diabetes at early stage using data mining techniques," in *Computer Vision and Machine Intelligence in Medical Image Analysis*. Springer, 2020, pp. 113–125.

[6] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive computation*, vol. 1, no. 2, pp. 139–159, 2009.

[7] L. Ge and K. K. Parhi, "Classification using hyperdimensional computing: A review," *IEEE Circuits and Systems Magazine*, vol. 20, no. 2, pp. 30–47, 2020.

[8] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, "Machine learning and data mining methods in diabetes research," *Computational and structural biotechnology journal*, vol. 15, pp. 104–116, 2017.

[9] B. Alić, L. Gurbeta, and A. Badnjević, "Machine learning techniques for classification of diabetes and cardiovascular diseases," in *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2017, pp. 1–4.

[10] A. Dagliati, S. Marini, L. Sacchi, G. Cogni, M. Teliti, V. Tibollo, P. De Cata, L. Chiovato, and R. Bellazzi, "Machine learning methods to predict diabetes complications," *Journal of diabetes science and technology*, vol. 12, no. 2, pp. 295–302, 2018.

[11] H. Naz and S. Ahuja, "Deep learning approach for diabetes prediction using pima indian dataset," *Journal of Diabetes & Metabolic Disorders*, vol. 19, no. 1, pp. 391–403, 2020.

[12] H. Kahramanli and N. Allahverdi, "Design of a hybrid system for the diabetes and heart diseases," *Expert systems with applications*, vol. 35, no. 1-2, pp. 82–89, 2008.

[13] G. A. Carpenter and N. Markuzon, "Artmap-ic and medical diagnosis: Instance counting and inconsistent cases," *Neural Networks*, vol. 11, no. 2, pp. 323–336, 1998.

[14] D. Deng and N. Kasabov, "On-line pattern analysis by evolving self-organizing maps," *Neurocomputing*, vol. 51, pp. 87–103, 2003.

[15] K. Kayaer, T. Yildirim *et al.*, "Medical diagnosis on pima indian diabetes using general regression neural networks," in *Proceedings of the international conference on artificial neural networks and neural information processing (ICANN/ICONIP)*, vol. 181, 2003, p. 184.

[16] J. C. Bioch, O. Van Der Meer, and R. Potharst, "Classification using bayesian neural nets," in *Proceedings of International Conference on Neural Networks (ICNN'96)*, vol. 3. IEEE, 1996, pp. 1488–1493.

[17] J. Park and D. W. Edington, "A sequential neural network model for diabetes prediction," *Artificial intelligence in medicine*, vol. 23, no. 3, pp. 277–293, 2001.

[18] I. Dankwa-Mullan, M. Rivo, M. Sepulveda, Y. Park, J. Snowdon, and K. Rhee, "Transforming diabetes care through artificial intelligence: the future is here," *Population health management*, vol. 22, no. 3, pp. 229–242, 2019.

[19] A. Rahimi, P. Kanerva, J. d. R. Millán, and J. M. Rabaey, "Hyperdimensional computing for noninvasive brain-computer interfaces: Blind and one-shot classification of eeg error-related potentials," in *10th EAI Int. Conf. on Bio-inspired Information and Communications Technologies*, no. CONF, 2017.

[20] A. Rahimi, P. Kanerva, L. Benini, and J. M. Rabaey, "Efficient biosignal processing using hyperdimensional computing: Network templates for combined learning and classification of exg signals," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 123–143, 2018.

[21] M. Imani, T. Nassar, A. Rahimi, and T. Rosing, "Hdna: Energy-efficient dna sequencing using hyperdimensional computing," in *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 2018, pp. 271–274.

[22] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[23] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[24] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[25] E. Fix and J. L. Hodges Jr, "Discriminatory analysis-nonparametric discrimination: Small sample performance," University of California, Berkeley, Tech. Rep., 1952.

[26] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, 2015.

[27] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.

[28] L. Bottou, "Online algorithms and stochastic approxima-p tions," *Online learning and neural networks*, 1998.

[29] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[31] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.

[32] J. Brownlee, "A gentle introduction to the rectified linear unit (relu)," *Machine learning mastery*, vol. 6, 2019.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[34] T. A. Hillier and K. L. Pedula, "Characteristics of an adult population with newly diagnosed type 2 diabetes: the relation of obesity and age of onset," *Diabetes care*, vol. 24, no. 9, pp. 1522–1527, 2001.

[35] D. S. Feig and V. A. Palda, "Type 2 diabetes in pregnancy: a growing concern," *The Lancet*, vol. 359, no. 9318, pp. 1690–1692, 2002.

[36] A. R. Frisancho, "Triceps skin fold and upper arm muscle size norms for assessment of nutritional status," *The American journal of clinical nutrition*, vol. 27, no. 10, pp. 1052–1058, 1974.
[37] J. G. Derraik, M. Rademaker, W. S. Cutfield, T. E. Pinto, S. Tregurtha, A. Faherty, J. M. Peart, P. L. Drury, and P. L. Hofman, "Effects of age, gender, bmi, and anatomical site on skin thickness in children and adults with diabetes," *PLoS One*, vol. 9, no. 1, p. e86637, 2014.
[38] Artem, *Diabetes predict score 0.92 and EDA*, Jupyter Notebook, 2021. [Online]. [Online]. Available: https://www.kaggle.com/kurazh/diabetes-prediction-score-0-92-and-eda
[39] D. Kleyko, A. Rahimi, D. A. Rachkovskij, E. Osipov, and J. M. Rabaey, "Classification and recall with binary hyperdimensional computing: Tradeoffs in choice of density and mapping characteristics," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 5880–5898, 2018.
[40] T.-T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Pattern Recognition*, vol. 48, no. 9, pp. 2839–2846, 2015.
[41] M. Imani, D. Kong, A. Rahimi, and T. Rosing, "Voicehd: Hyperdimensional computing for efficient speech recognition," in *2017 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE, 2017, pp. 1–8.
[42] J. Kim, H. Chang, D. Kim, D.-H. Jang, I. Park, and K. Kim, "Machine learning for prediction of septic shock at initial triage in emergency department," *Journal of critical care*, vol. 55, pp. 163–170, 2020.
[43] N. Watkinson, T. Givargis, V. Joe, A. Nicolau, and A. Veidenbaum, "Detecting covid-19 related pneumonia on ct scans using hyperdimensional computing," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 3970–3973.
[44] ——, "Class-modeling of septic shock with hyperdimensional computing," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 1653–1659.
[45] L. Denoyer and P. Gallinari, "Deep sequential neural network," *arXiv preprint arXiv:1410.0510*, 2014.
[46] B. M. Patil, R. C. Joshi, and D. Toshniwal, "Hybrid prediction model for type-2 diabetic patients," *Expert systems with applications*, vol. 37, no. 12, pp. 8102–8108, 2010.
[47] M. NirmalaDevi, S. A. alias Balamurugan, and U. Swathi, "An amalgam knn to predict diabetes mellitus," in *2013 IEEE international conference on emerging trends in computing, communication and nanotechnology (ICECCN)*. IEEE, 2013, pp. 691–695.
[48] E. Tartaglione, C. A. Barbano, C. Berzovini, M. Calandri, and M. Grangetto, "Unveiling covid-19 from chest x-ray with deep learning: a hurdles race with small data," *International Journal of Environmental Research and Public Health*, vol. 17, no. 18, p. 6933, 2020.
[49] A. D. Association *et al.*, "Diagnosis and classification of diabetes mellitus," *Diabetes care*, vol. 37, no. Supplement 1, pp. S81–S90, 2014.
[50] S. H. Wild and C. D. Byrne, "Risk factors for diabetes and coronary heart disease," *Bmj*, vol. 333, no. 7576, pp. 1009–1011, 2006.
[51] A. D. Association *et al.*, "The prevention or delay of type 2 diabetes," *Diabetes care*, vol. 25, no. 4, pp. 742–749, 2002.