IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), Bali, Indonesia, January 2021

Predicting Hospital No-Shows Using Machine Learning

Tasneem Batool, Mostafa Abuelnoor, Omar El Boutari, Fadi Aloul, Assim Sagahyroon
Department of Computer Science and Engineering
American University of Sharjah, UAE
{g00071143, b00069562, b00063858, faloul, asagahyroon}@aus.edu

Abstract—**All over the globe, significant amounts of patients miss their appointments without cancelling in time or even cancelling at all, resulting in billions of dollars wasted yearly due to increased idle time, overtime and waiting time that the other patients and hospitals face. Hospitals are actively trying to implement methods to try to reduce the idle time caused by patient no-shows by using overbooking and reminder systems. However, these two methods can be very costly. Overbooking can lead to patient dissatisfaction and constant personalized reminders, such as phone calls, to every patient can be annoying and costly in terms of manpower. This paper focuses on offering a solution which mitigates the global phenomenon of medical no-shows by creating a machine learning model using existing patient datasets to discover patterns and relationships between multiple patient variables and their tendency to miss appointments. Therefore, the likelihood of a patient showing up, given their information, may be predicted. The machine learning model used to form the solution predictive model is based on the decision tree classification algorithm. Furthermore, a scheduling system was implemented such that the overall model detects whether a patient has a risk of missing an appointment with a 95% accuracy, upon which it automatically enables the risky patient's schedule slot for overbooking and notifies medical staff or administration to contact them accordingly.**

Keywords – no-shows, machine learning, appointments

I. INTRODUCTION

A vast majority of clinics and hospitals around the world face a common major problem: huge sums of profit being wasted in the healthcare sector due to patient no-shows. In fact, it is estimated that around $150 billion is lost annually due to no-shows within the USA alone [1]. Any clinic that allows patients to book appointments in advance runs a huge risk; the patient not showing up to the appointment or cancelling the appointment without a sufficient time window. Hence many hospitals rely on the costly approach of *reminders* which is often implemented in parallel with *overbooking*, which is the practice of booking multiple patients in a single appointment slot. For reasons pertaining to the aforementioned economic and social implications, it becomes clear that a need exists for a solution which realistically tackles surging no-show rates and reduces the idle time and overtime of medical staff [2-3].

In this paper, an Appointment Scheduling and Intuitive Management (ASIM) system is proposed that enables quick, hassle-free appointment scheduling for patients and provides convenient and practical schedule oversight, logging, and management for the medical staff in a hospital or a clinic. The proposed solution allows for the automated handling of high-risk appointments, based on case-specific outcome predictions.

For the proposed ASIM system to predict a patient no-show, it was built to work in parallel with a machine learning classifier. Hence, to develop, test, and integrate a functional classifier, this paper utilizes a dataset which contains 110,527 appointment logs – recording appointment features such as Patient ID, gender, lead time, age, clinic neighborhood, health insurance, pre-existing medical condition, and a flag indicating whether or not the patient was reminded. By feeding the dataset's information into the classification algorithm selected, the system's machine learning component is trained sufficiently such that it is able to form rigid patient non-attendance predictions. As such, by using the prediction result evaluated by the classifier component, the system is able to overbook slots that will have a no-show. This paper explores and tests different, widely-used machine learning algorithms prior to integrating a classifier component capable of formulating predictions that are highly accurate and reliable such that the overarching system design may handle no-shows accordingly.

The remainder of the paper is organized as follows. Section II provides a literature review. Section III gives a description of the proposed system. Section IV discusses the implementation of the classifier. Section V explores the results of the classifier. Section VI evaluates the results obtained. The paper concludes in Section VII.

II. LITERATURE REVIEW

There have been significant efforts made by established researchers to investigate the phenomenon of hospital no-shows, predict its occurrences, and reduce its instances. Norris et al [4], Triemstra and Lowery [5], and McMullen and Netland [6] conducted research to map correlations between a set of independent variables and their impact on the rates of no-shows. Norris et al [4] distinguish themselves from their predecessors by considering three discrete outcomes on appointment attendance: show, no-show, and informed cancellation. This conscious decision prevents grouping the latter two outcomes into one and thus sharpens their focus on the problem of hospital no-shows. Moreover, while [5] and [6] have limited their independent variables to lead time (i.e. the time elapsed between the booking of the appointment and the appointment itself) and insurance, [4] have analyzed a wider range of independent factors which include weather, appointment time, lead time, prior attendance history, patient age, and payment method. All results from the analyses of [4-6] indicate that lead time plays the most important role in determining the attendance of a hospital appointment. It was observed that an increase in lead time led to an increase in the rates of hospital no-shows as well. [4] concluded their study by suggesting further research to determine the optimal lead time threshold that will improve appointment attendance. Similarly, [6] note that no-show rates are likely to decrease by as much as 60% if lead time were restricted to 0-2 weeks.

Aside from the surface analyses of data, researchers such as Mohammadi el al [7], Levy et al [8], and Alaeddini et al [9], have delved into the development of machine learning models to predict no-shows using a variety of algorithms. For instance, [7] created models using logistic regression, Naive Bayes, and artificial neural networks. Their dataset consisted of a vast array of independent variables such as clinic type, lead time, patient age, race, gender, marital status, cell-phone ownership, insurance, and tobacco usage. The resulting models reported accuracies of 73% for logistic regression, 71% for artificial neural networks, and 82% for Naive Bayes. Similarly, [8] also considered factors such as patient age, gender, and marital status, number of appointments that day, and patient diagnosis, producing a prediction model with an accuracy of 65%. [9] also worked with similar independent variables and produced a hybrid model of logistic regression and Naive Bayes, reporting an accuracy of 80%. Similar to [4-6], [7]'s models indicated that lead time once again played a significant role in the outcome of appointment attendance. [9]'s hybrid model on the other hand suggests a strong correlation between days close to holidays and no-shows, as well as types of clinics and no-shows.

On the aspect of no-show prevention, the two most popular methods observed in existing literature are overbooking and patient reminders. While overbooking does not directly address the problem of hospital no-shows, it is known to mitigate the consequences of no-shows, namely an increase in hospital idle time and decreased operational efficiency. Cao and Tang [10] and Chen [11] utilized different models for overbooking strategies in their works. [10] developed a Markov Decision Process (MDP) model to determine optimal overbooking strategy and proved the optimal overbooking policy is a threshold type policy; each appointment slot has a no-show probability and a threshold beyond which it is not optimal to book additional appointments on that slot. Similarly, [11] developed a simulation model to represent hospital appointments as a multi-server queue - each queue having its distinct overbooking strategy. [11]'s simulation results successfully improved the operational efficiency of the clinic by reducing overtime by 58%, idle time by 23%, and increasing the number of patients served by 16%.

While [10] and [11] have chosen to employ overbooking strategies, Walji and Zhang [12], and Percac-Lima et al [13] have chosen to research intervention systems such as patient reminders in an effort to reduce hospital no-shows and thus reduce wastage of resources. For instance, [12] uses human-computer interaction principles to test the type of messages that are more likely to appeal to patients and thus encourage them to attend their appointments on time. Through multiple iterations in their methodology, they were able to develop scripts for reminder emails that had a significant appeal to patients in terms of personalization, accuracy of the message, and an overall optimistic and "genuine" tone. While [13]'s study involved the development of classification models to predict no-shows much like [7-9], their research went a step further and employed an intervention method called the Patient Navigation System in an effort to minimize no-shows. [13] randomized and split their predicted no-show patients into a control group and intervention group of approximately the same size. Patients in the intervention group were called by trained callers to remind them of their appointments and resolve any barriers that they may experience. [13] were then able to conclude their study by stating that the intervention group reported a no-show rate 7.3% lower than that of the control group.

It is worth noting that [13], as well as Kaplan-Lewis and Percac-Lima [14], have also investigated the reasons for patient no-shows. The studies revealed that the main reasons for no-shows were forgetfulness and miscommunication. Miscommunication was further explained as patients being misinformed of the date and time of the appointment, patients incorrectly thinking that their appointments were cancelled, and patients being unaware of the existence of a cancellation process. This further strengthens the case for installing patient reminder systems as an intervention method to reduce hospital no-shows.

This paper seeks to contribute to the vast array of existing literature on two fronts. It was noted that the classification models developed by previous researchers reported accuracy measures between 65% to 82%; thus, one of the goals of this paper is to develop a classification model to predict hospital no-shows with an improved accuracy than that of its predecessors, paired with high precision and recall measures. Furthermore, this paper seeks to employ overbooking strategies limited to just those booked appointment slots which are predicted by the classifier to be no-shows, instead of a standardized, unadaptable mechanism including all patients. This will increase the overall operational efficiency of the hospital or clinic without imposing inconveniences on patients who show punctuality by attending their appointments on time.

III. Proposed System

The system proposed, as shown in Figure 1, consists of the following major components: a mobile application, web application, mobile app server, web app server, machine learning component, and a database. The system is implemented using the 3-tier architecture that contains the User Interface layer, Application Logic layer, and Database layer. The layered approach allows for decoupling, whose benefits include more manageable code, increased flexibility and ease in changing or upgrading parts of the implementation in the future, to name a few.

A. *User Interface Layer*

The User Interface consists of two client nodes, each node dedicated one type of client: either a patient client or a hospital admin client. Each client node consists of a component that allows the end user to interact with the system; the patient client node holds the mobile application component, whereas the admin client node holds the web application component. The patient accesses the mobile application to receive functionalities such as registering their profile and booking an appointment. The functionalities are achieved through the mobile application using the web interfaces provided by the

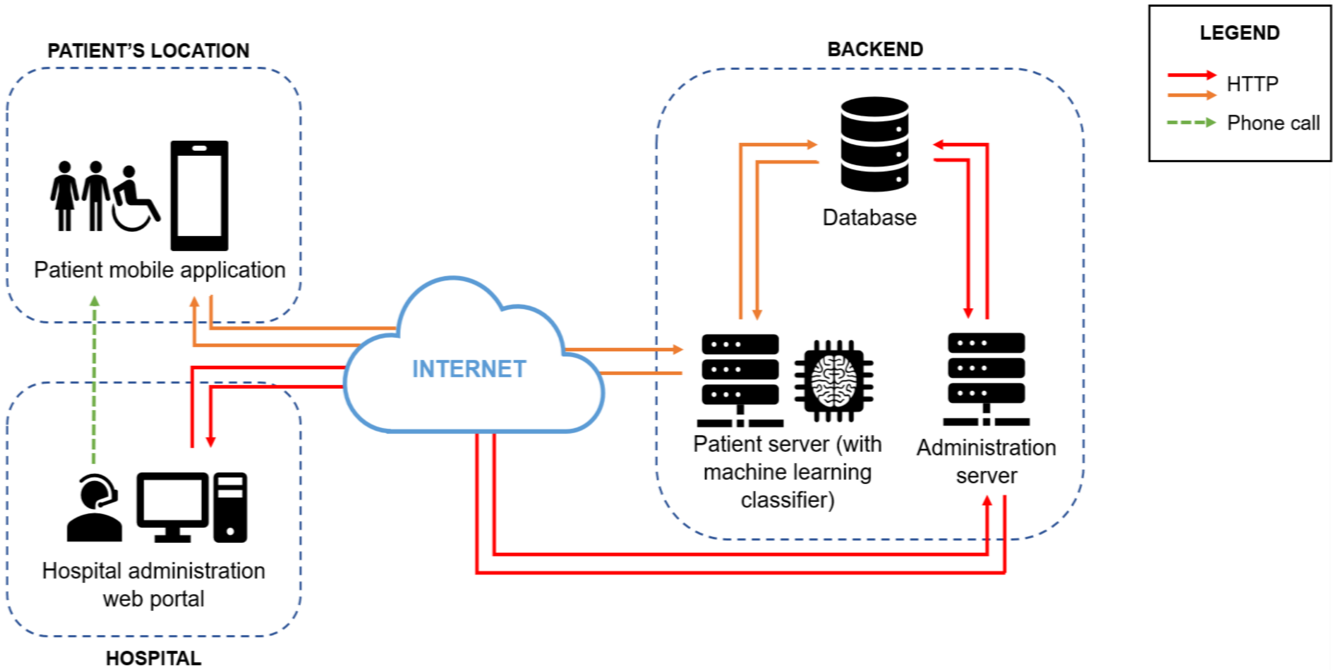

Fig. 1. System architecture for the proposed Appointment Scheduling and Intuitive Management (ASIM) System.

mobile app server. Similarly, the hospital admin accesses the web application to use functionalities such as adding appointment slots and viewing appointment schedules and details. These functionalities are once again achieved via the web application's use of web interfaces provided by the web app server.

B. *Application Logic Layer*

The Application Logic layer consists of two nodes, each node functioning as an application server to each type of client node. The mobile app server consists of 4 major components:

1) *HTTP server:* handles HTTP requests such, as GET and POST, from the mobile application. It provides the web interfaces for the mobile application to use to achieve its functionalities.

2) *Database service:* provides mediation between the server and the cloud database by providing web interfaces for the HTTP server to use, so it may achieve functionalities such as adding, modifying, or retrieving documents from the database.

3) *Classifier service:* provides mediation between the server and the trained machine learning classifier by providing the interface for the server to send patient and appointment information and receive the no-show prediction.

4) *Prediction classifier:* is the trained classifier used to predict the no-show likelihood of a patient, given their information and other additional metrics such as the number of days between the appointment day and the day the appointment was booked (lead time). The classifier service uses the prediction classifier to compute no-show and return the results to the HTTP server.

Meanwhile, the web app server node consists of two major modules:

1) *HTTP server:* handles HTTP requests from the web application and functions similarly to the HTTP server in the mobile app server node.

2) *Database service:* provides mediation between the HTTP server and the cloud database, similar to the database service module in the mobile app server node.

C. *Database Layer*

The Database layer consists of one (virtual) node where the system's cloud database is stored. The cloud database node contains the following components:

1) *Hospital database:* is the main overall database in the cloud database service cluster. It consists of all collections and records pertaining to hospital affairs only. It uses the interface provided by the patients, appointments, and admins collections to perform database operations such as adding, modifying, or retrieving a document.

IV. CLASSIFIER IMPLEMENTATION

To implement the ASIM system's prediction functionality, a public dataset of medical appointment records was subjected to various stages of data preprocessing, such as categorical data encoding and class balancing, prior to being used as input to numerous classification algorithms for early testing, performance evaluation, and, ultimately, model selection.

TABLE I. DETAILS OF HYPER-PARAMETERS TUNED FOR CLASSIFICATION.

Algorithm	Hyperparameter	Tuning Range
Decision Trees	min_samples_split	Small values in the range of 2 to 92 (inclusive), with increments of 10. Larger values in the range of 100 to 2000 (inclusive), with increments of 100.
Naïve Bayes	var_smoothing	Small values in the range of 10^{-9} to 10^{0} (i.e. 1, inclusive), with a multiplication factor of 10 per iteration. Larger values in the range of 5 to 100 (exclusive), with increments of 5.
K-Nearest Neighbor	n_neighbours	Values in the range of 1 to 30 (inclusive), with increments of 1.
Support Vector Machines (Linear)	C	[0.1, 0.5, 1, 5, 10, 15, 20, 25, 30, 40, 50, 75, 100]
	gamma	Defaulted to [1/(n_features * input variance)]
Support Vector Machines (Non-linear)	C	Defaulted to 1.0
	gamma	[0.001, 0.002, 0.003, 0.004, 0.008, 0.009, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1]

1) *Dataset used:* The public dataset used in this paper originates from Brazil. Published in May 2016, the dataset consists of a total of 110,527 records of hospital appointments, booked by patients, that are labeled by appointment outcome (i.e. show/no-show) [15]. The dataset consisted of 13 features, namely: Patient ID, Appointment ID, patient gender, day of scheduling, appointment day scheduled, patient age, patient neighborhood, health insurance, hypertension, diabetes, alcoholism, disability, and an “SMS reminder used” indicator.

Patient ID and Appointment ID held discrete numerical values. The gender attribute consisted of categorical values, with ‘F’ denoting “female” and ‘M’ denoting “male”. The appointment day and scheduling day attributes contained values that included the year, month, date, hour and minutes. The age attribute held discrete numerical values, with ages ranging from 0 to 115, with the exceptions of a few outliers (such as negative age). The neighborhood attribute consisted of categorical data with up to 80 distinct neighborhoods. The health insurance, hypertension, diabetes, alcoholism, disability, and SMS received attributes held Boolean values 0 and 1 (0 denoting negative, or false, and 1 denoting positive or true). The records in the dataset were labeled either “Yes” or “No”; where “Yes” signifies that the patient did indeed miss their appointments, and “No” meant that the patient attended his/her appointment. Initial data exploration revealed that of the 110,527 records, 22,319 (which makes up 20.19% of the dataset) were labeled “Yes”, while the rest were labeled “No”.

1) *Data pre-processing:* The data preprocessing stages included converting textual and categorical data into representative numerical values. In the example of the gender attribute, “F” was replaced by “1” and “M” was replaced by “0”; hence, in effect, the gender attribute was in essence a Boolean feature that indicated whether a given patient was female or not.

Similarly, the neighborhood attribute values were altered from names of the neighborhood to discrete numbers, where each number represented a distinct neighborhood. In addition, a new feature was created by finding the difference between the existing “Booking Day” and “Appointment Day” features. This new attribute, named “Lead time”, represents the number of days elapsed between the scheduling day and the appointment day. This was done such that the “lead time” attribute - the significance of which is established by previous literature - is explicitly taken into account by the model to be created. The team utilized the well-documented and easy-to-use Pandas library in Python to preprocess the original dataset.

2) *Data balancing:* Upon inspecting the percentage distribution of the records between the ‘Show’ and ‘No-show’ labels defined, it was observed that a large disparity between both classes exists, where 80% of the dataset’s records are labelled as ‘Show’ and 20% are labelled as ‘No-show’. Such a magnitude of class imbalance is to be expected for a dataset which logs information on patient attendance, which is a problem where the general trend of the majority of medical bookings is to be attended, as opposed to a minority of appointments which are not. In order to solve the problem of class imbalance in the used dataset, and after testing various undersampling and oversampling algorithms, the Instance Hardness Threshold (IHT) undersampling algorithm provided by the Imbalanced-learn (IMB-Learn) [16] library yielded the largest performance improvement when tested. Not only did IHT solve the issue of class imbalance, but this algorithm also improved the classification performed by a factor comfortably above 10% in all accuracy, precision, and recall measures.

3) *Classifier Implementation and Model Selection:* In this phase of model development and testing, and since the records retrieved from the Brazilian dataset are labelled for the outcomes of show/no-show, the subset of machine learning algorithms required to solve patient attendance prediction is that of classification - a form of supervised learning. Within the scope of this paper, since the model to be developed is to be classifying show/no-show, the classification process is synonymous with prediction. To develop supervised learning models which perform classification, this paper utilizes the robust and open-source Scikit-learn (SK-Learn) library, which implements various machine learning algorithms and model performance analysis functions using the Python programming language.

Within the scope of this paper, a classification model is defined as an algorithm – implemented by a pre-defined function from the SK-Learn library – which takes a distinct possible set of parameters as defined in the publicly available SK-Learn documentation [17]. For each model to be tested, the dataset was split into a training set, consisting of 80% of the total records, and a testing set, consisting of 20% of the total records, 10 times in a randomized manner. The average results across all 10 corresponding runs are reported. Table I details all the relevant hyper-parameters used for classification performance tuning. In addition to the relevant hyper-parameters, the table contains the range of hyper-parameter values selected, per algorithm, to test and fine-tune the different models explored.

Furthermore, the following list specifies the algorithms tested, using the SK-Learn API:

- Decision Trees (DT)
- Naïve Bayes – Gaussian (NB)
- K-Nearest Neighbors (KNN)
- Support Vector Machines (SVM) – Linear
- Support Vector Machines (SVM) – Non-linear

In this stage of model development, the dataset which was balanced by the IHT algorithm was used. Moreover, each algorithm was passed under an exhaustive hyper-parameter tuning technique, called grid search – implemented in the SK-learn library, that finds the set of optimal hyper-parameters given an algorithm, a parameter grid (i.e. a set of ranges for each parameter defined for that respective algorithm), and a scoring performance function to be used as a criteria for optimal parameter selection. The parameter grid used for grid search is retrieved from the parameter values in Table I. In addition, the number of features in the dataset is empirically reduced to the minimum number of features needed to obtain similar performance as that obtained using the complete, balanced dataset. The two forms of feature reduction tested include feature selection and dimensionality reduction – both of which also have functions implemented in the SK-learn library. Feature selection is a feature reduction technique which truncates the columns (i.e. features) within a dataset to keep the most important/informative K features (using a parameter also labeled 'K').

Dimensionality reduction, on the other hand, reduces the total number of dimensions in a sample set to a smaller number of variables, called principle components, which, to a limited extent, represent and summarize the information existing in the original features. To perform dimensionality reduction, this paper utilizes the Principal Component Analysis (PCA) function provided by SK-Learn, where the desired number of principle components obtained post-dimensionality-reduction is stored in a parameter labelled 'n_components'.

V. CLASSIFIER RESULTS

This section presents the results obtained after balancing the dataset classes using IHT under-sampling, performing feature reduction, and finding the optimal parameters per classifier by running a grid search on each. The results below illustrate the performance obtained when testing the models mentioned in Section IV of this paper via multiple, defined metrics.

In order to report on and evaluate model performance, this paper utilizes the following performance metrics:

1) *Accuracy:* a simple ratio of total correct predictions over total incorrect predictions.

2) *Precision:* (Also called positive predictive value) is the ratio of correctly predicted instances per class to all predictions made for that same class.

3) *Recall:* (Also known as sensitivity) is the ratio of the correctly predicted instances per class to the total amount of actual instances labelled to that class in the dataset.

4) *F1-measure:* a harmonic mean of recall and precision.

Moreover, in order to verify the predictive performance of the fine-tuned models reported below and judge performance consistency when subjected to a new data set (i.e. newly seen test data), each precision and recall measure obtained was compared to a set of exactly five respective precision and recall measures estimated through K-fold cross-validation (where K=5, in the case of this experiment); which is a method of a resampling a dataset by which the total data is split into K folds, after which, K model-testing iterations are created where each iteration represents a different fold used as a testing set, whilst all other folds are used for model training. In doing so, since such cross-validation ensures that the testing folds are diversified and that each subset within the data is used within a testing set at least once, the precision and recall performance of each classification model is empirically proven to perform well on unseen, limited data samples. Further, after performing 5-fold verification for each tuned model, the precision and recall measures obtained at each iteration of the cross-validation test was identical to the post-tuning measures reported in the sections below (with an error of +/- 1%).

A. Decision Trees

Without any feature reduction applied to the input dataset, the DT classifier yielded the averaged metrics shown in Table II.

TABLE II. CLASSIFIER PERFORMANCE WITH NO FEATURE SELECTION.

DT Performance (No feature reduction in dataset)				
Training Accuracy	Testing Accuracy	Precision	Recall	F1-measure
95.3%	94.5%	95%	95%	95%

The objective of feature reduction is to find the lowest values for K - the number of features to select using the SelectKBest algorithm, Percentile - the percentage of features to select using the SelectPercentile algorithm (which is similar to SelectKBest - except it uses percentage of features instead of number of reduced features K), or n_components – the number of components for dimensionality reduction which yield similar performance measures as the ones obtained with no feature reduction. According to the afore-described criteria, and as is demonstrated in the reported measures below, the values of 1, 10%, and 3 for K, Percentile, and n_components, respectively, are the best fit.

When K = 1, which is the lowest value possible for K, the DT classifier yielded the post-feature-selection metrics shown in Table III.

TABLE III. CLASSIFIER PERFORMANCE WITH K=1.

DT Performance (SelectKBest, K=1)				
Training Accuracy	Testing Accuracy	Precision	Recall	F1-measure
95.2%	95.4%	96%	95%	95%

When Percentile = 10%, which is the lowest value possible for 'Percentile', the DT classifier yielded the metrics post-feature-selection as shown in Table IV.

TABLE IV. CLASSIFIER PERFORMANCE WITH PERCENTILE = 0.1.

DT Performance (SelectPercentile, Percentile = 0.1)				
Training Accuracy	Testing Accuracy	Precision	Recall	F1-measure
95.3%	94.9%	95%	95%	95%

When n_components (using PCA) = 2, the DT classifier yielded the metrics post-dimensionality reduction as shown in Table V.

TABLE V. CLASSIFIER PERFORMANCE WITH N_COMPONENTS = 2.

DT Classifier Performance (PCA, n_components = 2)				
Training Accuracy	Testing Accuracy	Precision	Recall	F1-measure
63.3%	60.3%	60%	60%	60%

However, after increasing n_components to 3, the DT classifier yielded the metrics post-dimensionality reduction as shown in Table VI.

TABLE VI. CLASSIFIER PERFORMANCE WITH N_COMPONENTS = 3.

DT Classifier Performance (PCA, n_components = 3)				
Training Accuracy	Testing Accuracy	Precision	Recall	F1-measure
94.4%	94.5%	95%	95%	95%

After performing a grid search on the DT classifier, the optimal parameter value to maximize precision was found to be min_samples_split = 1400. However, min_samples_split = 2000 also returns a maximized identical precision value. The grid search algorithm returned 1400, which represents a more complex decision tree constructed by the DT model, compared to 2000, since the algorithm is designed such that the lowest value for a parameter is returned. In other words, the SKLearn grid search algorithm used was implemented such that a lower hyper-parameter value is assumed to represent a less costly model; which isn't the case with DT algorithm and min_samples_split.

B. *Other Prediction Models (NB, KNN, SVM)*

Using an identical criterion as the one explained in the previous section, and compared to the performance obtained for each model without applying any feature reduction, each of the NB, KNN, Linear SVM, and Non-Linear SVM classifier models was found to perform just as well post-feature-reduction using the values of 1, 10%, and 3 for K, Percentile, and n_components, respectively. The results are displayed in Table VII.

TABLE VII. CLASSIFIER RESULTS FOR ALL ALGORITHMS.

Algorithm	Accuracy	Precision	Recall	F1-measure
DT	94.5%	95%	95%	95%
NB	85%	80%	85%	85%
KNN	92.5%	90.5%	90.5%	90.5%
SVM (Linear)	92%	89%	88%	88%
SVM (Non-linear)	94.5%	95%	95%	95%

C. *Grid Search Results*

By performing a grid search on each of the classification models constructed and tested, the optimal combination of parameter values - retrieved from the hyper-parameter grid defined within the scope of this paper - for each model was found. Table VIII shows the grid search results for all hyper-parameter value sets which maximize precision, and other hyper-parameter sets which maximize recall.

TABLE VIII. GRID SEARCH RESULTS

Score Function	Precision	Recall
Decision Trees	min_samples_split = 2000	min_samples_split = 2
Naïve Bayes	var_smoothing = 0.1	var_smoothing = 100
KNN	n_neighbors = 2	n_neighbors = 1
SVM (Linear)	C = 0.1	C = 50
SVM (Non-linear)	gamma = 0.002	gamma = 0.1

VI. EVALUATION

As can be seen in the prediction results reported for all five classification models in Table VII, this paper manages to construct reasonably high-performing no-show prediction models given the dataset used; where some of the classifiers reached accuracy and f-measure scores both hovering the 95% benchmark. Moreover, both the DT and Non-linear SVM classification models were the best performers with identical performance reported via all average accuracy, average precision, average recall, and average F-measure metrics. However, since the DT algorithm has a significantly smaller time complexity with respect to that of the Non-linear SVM model, the DT classifier was selected to be integrated into the proposed ASIM system design documented by this report.

Furthermore, to confirm whether the review of literature, which emphasized lead time, was accurate, Table IX shows the estimated performance measures of the DT classifier when the input dataset contains the lead time attribute compared to its performance when the lead time attribute is omitted from the dataset. The differences exhibited in performance with and

TABLE IX. LEAD TIME VS. NO LEAD TIME

Attribute	Lead time	No Lead time
Avg. Training Accuracy	95.26%	60.13%
Avg. Testing Accuracy	95.28%	59.24%
Avg. Precision	92.60%	58.23%
Avg. Recall	98.37%	65.59%
Avg. F1-measure	95.39%	61.59%
Avg. True Positives	4411.9	2925.3
Avg. False Negatives	73.3	1536
Avg. False Positives	352.7	2102.3
Avg. True Negatives	4089.1	2363.4

without the lead time attribute in the input dataset by the DT classifier is significant.

With lead time, the DT classifier was able to obtain performance measures comfortably above 92% in all measures used. However, after running the exact same model with the lead-time attribute eliminated from the input dataset, a drop of over 30% was observed in all classification performance metrics yielded. After performing the same test for the NB, KNN, Linear SVM, and Non-linear SVM models, a similar drop in performance was observed. This observation confirms a key literature review finding that most relevant works consider, emphasize, or even explicitly list lead-time as a highly informative determinant variable when analyzing patient non-attendance or formulating a no-show prediction [4-9].

VII. CONCLUSION

In this work, the Appointment Scheduling and Intuitive Management (ASIM) system was able to detect patient no-shows using a machine learning classifier. This paper was able to achieve a highly rigid prediction performance rate with a score of 95% for both accuracy and F1-score measures.

As the paper outlines, various algorithms were used to build and test the multiple models reported. The models built were tested using the dataset, after which different metrics such as accuracy, precision and f-measure were used to compare the results of the different algorithms and classifiers built. As shown earlier, in order to get optimal results from each algorithm, different parameters were tweaked and tested via multiple grid search runs. It was also observed that the most determining factor in obtaining an accurate result was the existence of the lead time attribute in the dataset, without which classifier performance - resembled in accuracy, for instance - drops by an astonishing 30%. After all the tests, the best accuracy and F1-measure achieved for the dataset used in this paper was that of 95% for both. This was accomplished by using the decision trees classifier, with the parameter "min_samples_split" set at 2000.

Ultimately, the system proposed in this paper allows for a reduction of financial losses globally due to no-shows. By deploying the ASIM system and further improving its design, clinics can now reliably predict a no-show and, in turn, overbook or remind the patient in question to minimize opportunity cost. The proposed system may also be extended to other businesses that have a high chance of customer congestion, cancellation possibility, and inherent financial losses per booking missed, such as hotels and restaurants.

REFERENCES

[1] J. Gier, "Missed appointments cost the U.S. healthcare system $150B each year," *Clinical IT*, April 2017.

[2] S. Dhal, "How patient no shows have a ripple effect on clinics, hospitals," *Gulf News*, April 2019.

[3] "Missed GP appointments costing NHS millions," *National Health Service*, January 2019.

[4] J. Norris, C. Kumar, S. Chand, H. Moskowitz, S. Shade, and D. Willis, "An empirical investigation into factors affecting patient cancellations and no-shows at outpatient clinics," *Decision Support Systems*, vol. 57, 428-443, January 2014.

[5] J. Triemstra and L. Lowery, "Prevalence, Predictors, and the Financial Impact of Missed Appointments in an Academic Adolescent Clinic," *Cureus*, vol. 10, no. 11, e3613, November 2018.

[6] M. McMullen and P. Netland, "Lead time for appointment and the no-show rate in an ophthalmology clinic," *Clinical ophthalmology*, vol. 9, 513-516, March 2015.

[7] I. Mohammadi, H. Wu, A. Turkcan, T. Toscos, and B. Doebbeling, "Data Analytics and Modeling for Appointment No-show in Community Health Centers," *Journal of primary care & community health*, vol. 9, Jan-Dec 2018.

[8] V. Levy, "A predictive tool for nonattendance at a specialty clinic: An application of multivariate probabilistic big data analytics," in Proc. of the *10th International Conference and Expo on Emerging Technologies for a Smarter World* (CEWIT), October 2013.

[9] A. Alaeddini, K. Yang, C. Reddy, and S. Yu, "A probabilistic model for predicting the probability of no-show in hospital appointments," *Health care management science*, 14(2), 146-157, June 2011.

[10] P. Cao and J. Tang, "Hybrid appointment for a single-physician clinic with no-shows and overbooking," in Proc. of the *IEEE Chinese Control and Decision Conference* (CCDC), June 2014.

[11] J. Chen, "Show or No Show? Visualization and Prediction," *Kaggle.com*, 2020. [Online]. Available: *https://www.kaggle.com/jchen2186/show-or-no-show-visualization-and-prediction*.

[12] M. Walji and J. Zhang, "Human-Centered Design of Persuasive Appointment Reminders," in Proc. of the *IEEE Annual Hawaii International Conference on System Sciences* (HICSS), January 2008.

[13] S. Percac-Lima, P. Cronin, D. Ryan, B. Chabner, E. Daly, and A. Kimball, "Patient navigation based on predictive modeling decreases no-show rates in cancer care," *Cancer*, 121(10), 1662-1670, May 2015.

[14] E. Kaplan-Lewis and S. Percac-Lima, "No-Show to Primary Care Appointments: Why Patients Do Not Come," *Journal of Primary Care & Community Health*, 4(4), 251-255, October 2013.

[15] JoniHoppen, "Medical Appointment No Shows," *Kaggle*, 20-Aug-2017. Available: https://www.kaggle.com/joniarroba/noshowappointments

[16] "Welcome to imbalanced-learn documentation!" *imbalanced*. [Online]. Available: https://imbalanced-learn.readthedocs.io/en/stable/index.html

[17] Pedregosa et al. "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, 2825-2830, October 2011.