IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 29, 2021, accepted June 18, 2021, date of publication June 22, 2021, date of current version July 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3091622

ICU Survival Prediction Incorporating Test-Time Augmentation to Improve the Accuracy of Ensemble-Based Models

SEFFI COHEN[1], NOA DAGAN[1,2,3], NURIT COHEN-INGER[4], DAN OFER[5], AND LIOR ROKACH[1]
[1]Department of Software and Information Systems Engineering, Ben-Gurion University, Beer-Sheva 8410501, Israel
[2]Clalit Research Institute, Clalit Health Services, Tel Aviv 44457, Israel
[3]Department of Biomedical Informatics, Harvard Medical School, Boston, MA 02115, USA
[4]BeyondMinds, Tel-Aviv 5800001, Israel
[5]Medtronic, Tel-Aviv 5912300, Israel

Corresponding author: Seffi Cohen (seffi@post.bgu.ac.il)

ABSTRACT This work presents a novel method for applying test-time augmentation (TTA) to tabular data. We used TTA along with an ensemble of 42 models to achieve higher performance on the MIT Global Open Source Severity of Illness Score dataset consisting of 131,051 ICU visits and outcomes. This method achieved an AUC of 0.915 on the private test set (19,669 admissions) and won first place at Stanford University's WiDS Datathon 2020 challenge on Kaggle, while the Acute Physiology and Chronic Health Evaluation (APACHE) IV model (commonly used for ICU survival prediction in the literature) achieved an AUC of 0.868. In addition to increasing the AUC score, our method also reduces "unfair" bias.

INDEX TERMS Ensemble methods, machine learning, supervised classification, healthcare.

I. INTRODUCTION

There are many score systems based on clinical data that successfully measure survivability of critically ill patients. One widely used system is the Acute Physiology and Chronic Health Evaluation (APACHE) IV, which is well-known as a performance benchmarking system in ICUs [2]. APACHE IV is based on a logistic regression model, developed on a dataset of over 131,000 ICU admissions. The data includes age, gender, hospitalization conditions, physiologic data, and chronic health conditions (such as hepatic failure, immunosuppression, lymphoma, leukemia, etc.). Stanford University's Women in Data Science (WiDS) initiative suggested that the Kaggle community address the challenge of ICU survival prediction [3], based on MIT's Global Open Source Severity of Illness Score (GOSSIS) dataset [4]. Our method for this challenge was based on an intensive preprocessing of the data, and an ensemble of 42 different models, including K-nearest neighbor, gradient boosted trees, random forest, neural networks, and logistic regressions. We created the models' ensemble by applying three layers of the StackNet meta-learning architecture [5]; finally, for the creation of the assigned predictions we also augmented the test set by generating three similar samples for each record, each time by altering the value of either the gender, ethnicity or age variables. The final predictions were created by averaging the probabilities assigned to each of the three duplicate records. Figure 1 demonstrates our method pipeline (from left to right).

The contributions of this paper are twofold. First, we demonstrate a novel application of the test-time data augmentation method to tabular data, both for increasing the model generalizability and for debiasing potentially discriminating variables such as age, gender, and ethnicity. Second, we present a new machine learning-based ICU calculator that outperforms existing models and was ranked first place in the 2020 Stanford WiDS competition.

The associate editor coordinating the review of this manuscript and approving it for publication was Hiu Yung Wong.

II. RELATED WORK

A. PATIENT SURVIVAL MODELS

APACHE IV [2] is one of the well-known scoring systems for predicting hospital mortality among critically ill adults. APACHE IV significantly improved upon previous scoring systems, such as APACHE III [9], Simplified Acute Physiology Score (SAPS) II [10], and the Mortality Probability Model (MPM) II [11]. APACHE IV uses a multivariate

FIGURE 1. **Survival Prediction Incorporating Test-Time Augmentation. a - For each patient, the dataset contains 185 features. b - fifty percent of the features had more than 50% missing values; we handled the missing values by using predictive imputation and distribution-based approaches. c - Using several different techniques of feature selection, we eliminated features that were not useful. d - We generated 42 models to ensemble different points of view. e - To maximize the generalizability of the ensemble, we applied three layers of the stacking method. f - We augmented the test set by generating new records. g - The average of the augmented test set served as the survival prediction.**

logistic regression model and data from the first 24 hours of an ICU admission, including demographics, vital signs, blood gases and laboratory test results, urine output, Glasgow Coma Scale, ICU admission diagnosis and source, and history of chronic diseases. This score is also widely used as the state-of-the-art benchmark for ICU mortality prediction [12], [13]. In addition to APACHE IV, various machine learning methods have also successfully been applied for ICU survival prediction, including non-negative matrix factorization [36], neural networks [38], and domain adaptation [39]. In some cases, the machine learning approaches showed better predictive performance than the scoring systems mentioned above [37].

B. DATA AUGMENTATION AND TEST-TIME AUGMENTATION

A small number of training samples may lead to overfitting. One way of increasing the size of the training set is to generate artificial data by augmenting the original training data [6]. Data augmentation is widely used for computer vision tasks, especially for convolutional neural networks. Common data augmentation methods used for image augmentations include

scaling, translation, rotation, random cropping, image flipping and color shifting.

In addition to augmenting training data, augmenting images at test time (TTA) has had good results [7], and it is commonly used in medical image diagnosis, such as skin lesion analysis [8]. The diversity principle of ensemble methods served as the inspiration for test-time augmentation, in the data space. TTA is accomplished by taking a test image, augmenting it, and then averaging the predicted probabilities of the augmented images. Unlike image augmentation, there is no existing TTA technique for tabular data, and the novelty of our work is to apply TTA on such data.

III. PROBLEM FORMULATION

We frame the problem of predicting patient survival as a classification problem. Given a new ICU patient, the goal is to predict the probability that the patient's stay in the ICU will culminate in death. We use all available data from the patient's first 24 hours in the ICU as features for our model. Given a database $D^{n\times m}$, and a labels vector y^n, where n is the number of admissions in the dataset. Each ICU admission is described by a set of m attributes collected from the first 24 hours of the patient's stay, and y indicates whether the patient dies.

We optimized the classifier C by minimizing the log loss:

$$-\frac{1}{n}\sum_{i}^{n} y_i \log p_i \tag{1}$$

where p_i is the probability the classifier assigned to label i.

A. DATA

MIT's international GOSSIS [4] consortium has put together various databases to create a resource that allows assessment of illness severity. GOSSIS provides a dataset of 131,051 ICU admissions, spanning a one-year timeframe and contains patient demographics data, lab results, chronic health conditions, APACHE scores and various medical measures, including the minimum and maximum values for the first hour and first 24 hours of the ICU admission, as illustrated in Figure 2. Overall, the provided dataset included 185 features. The target variable to predict was the death outcome in the ICU. The training set was unbalanced, with 7,915 deaths out of the total 91,713 admissions. The dataset is available at https://www.kaggle.com/c/widsdatathon2020/data

IV. METHODS

In this section, we describe the main part of our work - the TTA method, data imputation, feature selection and modeling.

A. TEST-TIME AUGMENTATION

In addition to increasing the generalizability and reducing the error bias of the final model, our TTA approach can also reduce "unfair" bias. We augmented the test set by creating three additional records, in each case altering the value of either the gender, ethnicity, or age variables. The ensemble

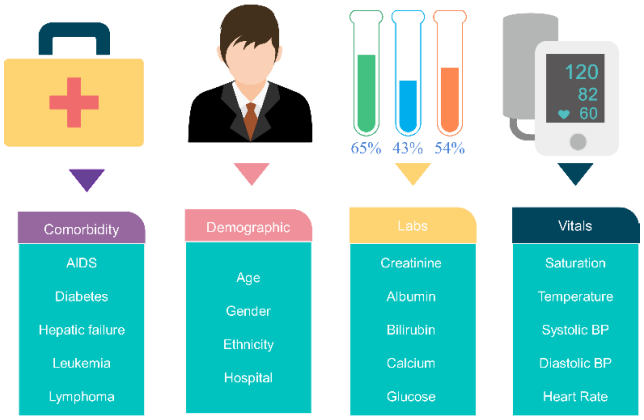


FIGURE 2. The dataset contains 185 features, including patient demographic data, lab results, chronic health conditions, APACHE scores, medical indices, and statistics on the patient's first 24 hours in the ICU.

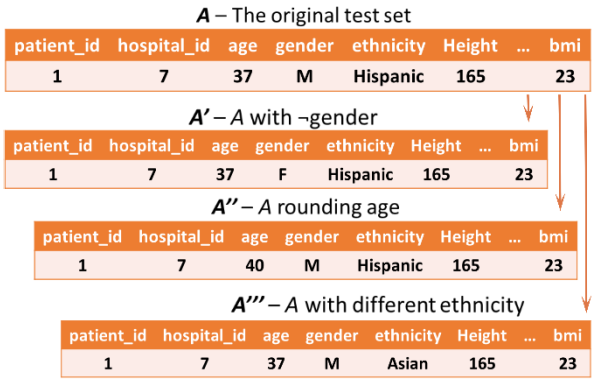


FIGURE 3. Example of the TTA method on one patient. For each patient, we produced additional three records. Then, we predict the augmented records and average the predictions to create the final prediction.

methods with diverse models, combined with the TTA technique, were all used to increase accuracy and reduce overfitting.

M is our model, and A is our original test set. Let A_i be the set of all variables selected for augmentation. For example, the age attribute on A is A_{age}, the gender attribute is A_{gender}, and the ethnicity is $A_{ethnicity}$. We augmented the test set by generating three new test sets, based on A as the following:

- A' is similar to A, where $A'_{gender} = \neg A_{gender}$.
- A'' is similar to A, where A_{age} rounded as follows:

$$A''_{age} = Round\left(\frac{A_{age}}{10}\right)10 \tag{2}$$

- A''' is similar to A, after setting $A_{ethnicity}$ values uniformly at random.

After the production of the augmented datasets, we predicted the likelihood of death by averaging all predicted probabilities as follows:

$$TTA_{prob} = \frac{M(A) + M(A') + M(A'') + M(A''')}{4} \tag{3}$$

TTA prediction examples are presented in Figure 3.

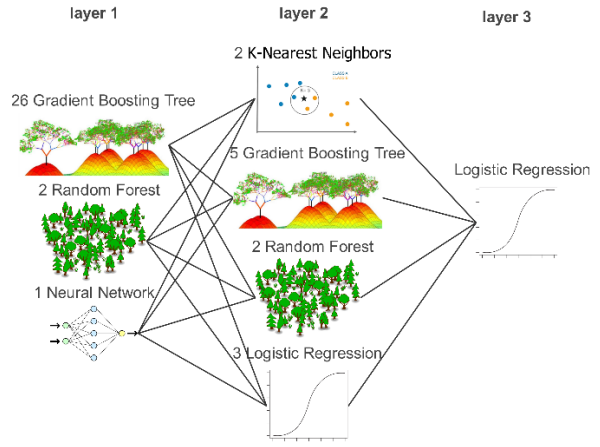


FIGURE 4. Our architecture is based on an ensemble of 42 different models, including K-nearest neighbor, gradient boosted trees, random forest, neural networks, and logistic regression.

B. THE ENSEMBLE ARCHITECTURE

The ensemble method we used is based on the StackNet [5] ensemble method. The StackNet model is a powerful ensemble method that combines the idea of meta-learning and feed-forward neural networks. The building blocks of this method are various machine learning models, each of which is a node in a modeling architecture of various layers. Each layer uses the predictions of the previous layer as input, except for the first layer, which uses the training set as input. We tried to maximize the model generalization by focusing on the diversity generator and the model's meta-learner as a combiner. We implemented the StackNet model using a stacking model, which are three layers deep, as shown in Figure 4, using a diverse combination of models.

Deep stacking is based on a complex ensemble of models, inspired by a fully connected neural network architecture. The first layer obtains the training set as input; then, the second layer obtains the first layer's predictions as input; and finally, the third layer obtains the second layer's predictions as input. In layer one, we used 26 gradient boosted trees models [17], two random forest models [18], and one artificial deep neural network. In layer two, we trained another 12 models. Finally, in layer three, we trained a meta-classifier based on the predictions produced by layer two. The meta-classifier algorithm used a logistic regression with brute force feature selection [19].

From a model diversity perspective, we trained the models with extremely different parameters, as described in Table 1.

C. DATA IMPUTATION

The main idea of imputation is that if an important feature is missing for an observation, it can be estimated from the data that is present. There are two main imputation approach families: predictive value imputation and distribution-based imputation [21]. In the given scenario, the data had a lot of missing values - more than 50% of the features had more than 50% missing values. In medical risk stratification tools, it is extremely important to use most of the data we had, rather than discarding observations or features with a high percentage of missing values. Since neither imputation method works for all the features, we used both methods as follows:

TABLE 1. The algorithms used in the deep stacker layers.

#	Level	Algorithm	Hyperparameters
1	1	Random Forest	Trees 100; depth 50
2	1	Random Forest	Trees 3000; depth 12
3-28	1	GBM	Different depth, learning rate, max_leaves, ignored_features, min_data_in_leaf
29	1	Deep Learning	Three layers, ReLU activation, Adam optimizer, 0.45 dropout between layers
30-32	2	Logistic Regression	Different feature selection methods
33-37	2	GBM	Different depth
38-39	2	KNN	Neighbors 100, neighbors 500
40-41	2	Random Forest	Different depth
42	3	Logistic Regression	Exhaustive brute force feature selection

1) PREDICTIVE VALUE IMPUTATION

- *Predictive model imputation* [35] - using linear regression to predict the feature using other features, e.g., height can be implied from weight, ethnicity, age and gender.
- Deducing probable values from other variables (based on domain knowledge), e.g., deducing ventilation status from the values of other relevant variables.

2) DISTRIBUTION-BASED IMPUTATION

- Statistical imputation - mean/median according to the feature distribution, e.g., the glucose in the blood can be imputed by the mean of this feature.
- Using the frequent value when the distribution is skewed, e.g., AIDS condition.

3) CATEGORIZATION

- Using domain knowledge to categorize numeric variables to meaningful bins and adding a missing category, since missingness often has an important predictive signal in medicine.

Our methodology consisted of iteratively imputing features and examining the improvement in the validation test. Although many of the models used supported missing values inherently (LightGBM, XGBoost, CatBoost) [22], we found it highly beneficial to impute missing values using different methods. As a result of the iterations and the methodology used, the test score for the predictions improved the AUC score by ∼0.003 from ∼0.906 to ∼0.909 using a single model of gradient boosted trees.

D. FEATURE SELECTION

We used several different techniques for feature selection. We evaluated these techniques individually, performing different experiments while iteratively dropping features in a fast pipeline, based on feature selection techniques that improved the validation set AUC score. We dropped features based on the following criteria:

- Features with a high percentage of missing values (more than 80%).
- Collinear features - with a threshold above 0.99.
- Features with zero standard deviation (only the 'readmission_status' feature met this criterion).
- Features with a high recursive feature elimination (RFE) ranking [32] (only the 'paco2_for_ph_apach' feature meet this criterion).
- Features with zero SHAP values. [33]
- According to adversarial validation models and Kolmogorov-Smirnov [34] tests, features with different distribution between the train and test sets were dropped (the 'icu_id' and the 'hospital_id' features met this criterion).
- Extremely important and "aggressive" features were dropped in most of the ensemble models (25 of the GBM and one random forest). We found that this improved our validation performance and test score; this was likely due to using the same features used by APACHE IV to calculate survival likelihood, which led the model to overfit with it - (features that met this criterion included 'apache_4a_hospital_death_prob', the APACHE IV probabilistic prediction of in-hospital mortality for the patient, and 'apache_4a_icu_death_prob', the APACHE IV probabilistic prediction of in ICU mortality for the patient).
- Adversarial validation – we checked the similarity between the training and test sets in terms of feature distribution, and then selected features with near-perfect train/test separation - with no coverage over the test set (the 'encounter_id', 'icu_id' and the 'hospital_id' features met this criterion).

V. EXPERIMENTAL STUDY

We performed an experimental study to evaluate our method and compare its performance, with TTA and without TTA, to the APACHE IV scoring system, a deep learning model, and an AutoML model, using the GOSSIS datasets. We also compared our method to the methods proposed by the five runners up in the WiDS Datathon 2020 Kaggle competition, specifically the five methods that followed us to the top of the private leaderboard (Nullset, Prevision.io, ML Keksika, EFPL, Dihydrogen Oxide).

A. EVALUATION

The evaluation metric used was the area under the receiver operating characteristic curve (AUC) [15] between the predicted mortality and the ground truth.

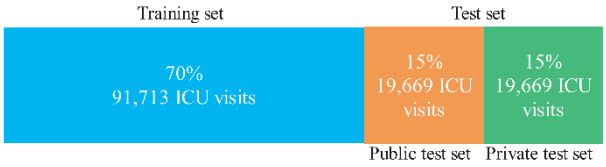


FIGURE 5. The MIT's GOSSIS dataset was split into a training set, a public test set, and a private test set.

The training-test set split was 70% for the training set and 30% for the test set. The test set was split into two equal sets for the public and private leaderboard, as illustrated in Figure 5. Public and private test sets – both are unseen data sets for the model's evaluation. The public test set is intended for validation during the competition. The evaluation of the private test set is announced only after the competition period ends to prevent overfitting of the test set.

B. BENCHMARK MODELS

Here we describe the methods and works we compared to our method. The quantitative results of the compared methods are presented in section 6.

1) APACHE IV

The latest *APACHE* scoring system for ICU survival prediction.

2) DEEP LEARNING

A basic TensorFlow 2.0 feedforward, fully connected, two-layer neural network, implemented and published by Rengaraju [23].

3) H2O AutoML

An automated platform for machine learning. H2O AutoML includes automatic training and tuning of deep learning, random forest, logistic regression, gradient boosting trees, and the stacking ensemble method. For the current dataset, AutoML builds the following models: random forest, extremely randomized forest, a random grid of generalized linear models (logistic regression with regularization search), deep learning, two types of gradient boosting trees (XGBoost and H2O GBM), a feedforward neural network, and two stacked ensembles - one with all of these models and one with only the best models of each kind; implemented and published by Rengaraju [24].

4) NULLSET (WiDS DATATHON COMPETING TEAM)

Ensemble method, based on 26 models, including XGBoost, LightGBM, and logistic regression [25].

5) PREVISION.IO (WiDS DATATHON COMPETING TEAM)

Ensemble of two LightGBM models and a neural network model was implemented, using a pseudo labeling technique [26], [27].

6) ML KEKSIKA (WiDS DATATHON COMPETING TEAM)

Ensemble of LightGBM models, with and without pseudo labeling technique, and a neural network model [28].

7) EFPL, (WiDS DATATHON COMPETING TEAM)

Ensemble of LightGBM models using the GOSS (Gradient-Based One Side Sampling) method [29], [30].

8) DIHYDROGEN OXIDE (WiDS DATATHON COMPETING TEAM)

A stacked ensemble of three gradient boosting trees, two XGBoost models with a LightGBM model [31].

VI. RESULTS

In this section, we present the quantitative results of the compared methods on MIT's GOSSIS ICU dataset.

Our method outperformed all other methods that were evaluated in this study. Other top performers include the APACHE IV, which achieved an AUC score of 0.868, a simple feedforward neural network which obtain an AUC score of 0.895, and the H2O AutoML platform, which built an ensemble of models, with an AUC score of 0.900. We also performed ablation testing to evaluate the TTA's contribution to our final predictor. Without TTA, our method achieved an AUC of 0.911, in contrast to an AUC of 0.915 with TTA. Table 2 presents a comparison of the compared methods performance.

TABLE 2. Comparison of APACHE IV, deep learning model, H2O AutoML, StackNet, and StackNet with TTA on MIT's GOSSIS ICU dataset.

#	Scoring System	Private AUC Score (SD)
1	APACHE IV	0.868 (0.0130)
2	Deep Learning	0.895 (0.0009)
3	H2O AutoML	0.900 (0.0018)
4	Our method without TTA	0.911 (0.0006)
5	**Our method**	**0.915 (0.0005)**

TABLE 3. Comparison of the top six methods on Kaggle's leaderboard.

#	Team Name	Public AUC Score	Private AUC Score (SD)
1	**Our method (Women Power)**	**0.9160**	**0.9150 (0.0005)**
2	Nullset	0.9138	0.9149 (0.0006)
3	Prevision.io	**0.9160**	0.9145 (0.0008)
4	ML Keksika	0.9156	0.9143 (0.0007)
5	EPFL	0.9149	0.9141 (0.0004)
6	Dihydrogen Oxide	0.9141	0.9140 (0.0001)

We also compared our method's performance to the performance of other top performing methods in the Kaggle competition, specifically the five runners up in the private leaderboard [20]. All these works implemented ensemble methods, and most of them also included stacked gradient boosting trees with neural networks. A comparison of our method's (''Women Power'') performance to that of the other top performers is presented in Table 3.

It is noted that our method, without the TTA, achieved almost the same results as the top-ranked ensemble methods in the competition. The TTA technique, when used along with the ensemble, provided additional accuracy and generalizability (i.e. maintaining the high accuracy on an unseen test set).

For all the top-ranked methods, there was a slight difference AUC scores between the public and private leaderboard; The standard deviation between the our private and public test sets was 0.0005. A large difference between the public and private leaderboard scores could indicate overfitting to the dataset of the public leaderboard. Figure 6 shows the comparison between the performance of each model in the public and private leaderboards. The leaderboard is available online from Kaggle [20].

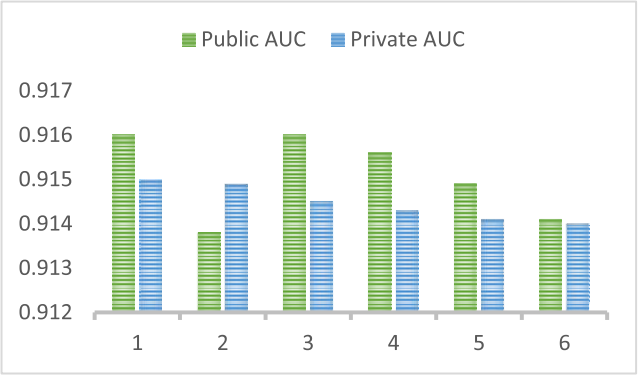


FIGURE 6. Public versus private leaderboard score. A large difference between the performance on the public leaderboard and the private leaderboard could indicate overfitting to the public leaderboard data. Our method achieved a high rank on both the public and private leaderboards. (Note that the numbers on the x-axis of the graph correspond to the method numbers in Table 3).

We also examined our method on a similar dataset for diabetes Mellitus classification [1]. This dataset is also provided by GOSSIS on WiDS 2021 and contains almost the same features but a different prediction target. Our method was ranked in the first place, on the first try.

We used SHapley Additive exPlanations (SHAP) [33] to assess the impact of the different features. SHAP is a game-theoretic solution concept which estimates the contribution of features on the model performance by comparing what a model predicts with and without any feature in the dataset. The Shapley values present the contribution of each feature to the model.

To compute the effect of feature i, the SHAP values φ_i is calculated as following:

$$\varphi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}\left(x_{S \cup \{i\}}\right) - f_S(x_S) \right] \tag{4}$$

- F represents the set of all features
- $S \subseteq F$ is all feature subsets
- f_S is a model trained on subset S
- $f_{S \cup \{i\}}$ is a model trained on subset S and feature i
- x_S represents the values of the input features in subset S

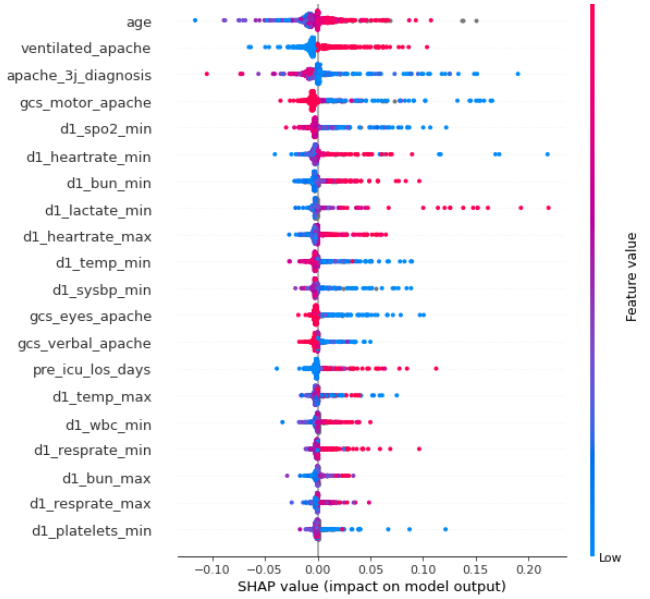


FIGURE 7. The top 20 SHAP values on the entire ensemble of models. The points on this plot represent the impact of the features on the prediction. For example, high values of 'age' caused higher probabilities, and low values caused low probabilities.

The top 20 features with the highest SHAP values on the entire ensemble of models are shown in Figure 7. The features are ranked in descending order of feature importance. The ''SHAP Value (Impact on model)'' is the horizontal location that shows whether the effect of that value is associated with a higher or lower prediction. The color shows whether that variable's value is high (in red) or low (in blue) for each observation.

It is noted that the most impactful feature was the patient's age; Consistent with clinical expectations, a higher age (red points) is associated with an increased probability of death (positive SHAP values), while a younger age (blue points) is associated with a decreased probability of death (negative SHAP values). The second feature in terms of importance was the ''ventilated_apache'' feature, a binary variable indicating whether the patient was mechanically ventilated or not; It is noticeable that ventilation increased the predicted mortality risk for all individuals (all the red dots, indicating a value of 1, have positive SHAP values), as expected based on clinical knowledge. The SHAP results for the different Glasgow Coma Scale (GCS) variables, indicating a person's level of consciousness using three sub-scores (eye, verbal, and motor), are also consistent with clinical expectations: low-performance scores (blue points) are located on the positive side of the SHAP contribution scale, indicating that individuals with these scores received higher predicted risk scores. The ''d1_spo2_min'' feature represents the minimal oxygen saturation measured on the first day of the ICU admission. Indeed, low oxygen values were found to increase the predicted risk. Another vital sign, heart rate (represented by the ''d1_heartrate_min'' feature that indicates the minimal heart rate measured on the first day), presents an interesting trend; both increased heart rates (tachycardia) and extremely low heart rates (bradycardia) increased the predicted risk. This demonstrates the strength of the final predictor in identifying non-linear patterns in variable effects on the studied outcome. Finally, the SHAP values for the various laboratory features are also presented in Figure 7. These values are consistent with clinical expectations.

VII. CONCLUSION

In this paper, we explored the effects of deep stacking ensemble methods and TTA using simple transformations on ICU survival prediction. Our evaluation showed that the use of this ensemble approach, combined with TTA, helped improve the accuracy of ICU survival prediction, achieving state-of-the-art results, and outperforming the widely used APACHE IV scoring system. This method's advantage was also demonstrated in the WiDS Datathon 2020 challenge, where it achieved first place (out of 951 teams) in a Kaggle competition.

The main limitation of our work results from constraints that have arisen from the fact that it was created as part of a Kaggle competition. Our goal was to create the most accurate model according to the evaluation metric defined by Kaggle (AUC), while there are additional important evaluation metrics that are relevant for clinical use. Another limitation is the model's lack of interpretability, which was compensated to some extent by the SHAP values analysis.

Our comparison of various methods showed that other top-ranked models were also based on an ensemble of neural networks and gradient boosting trees, but lacked the addition of the TTA. The AutoML platform, which was used as another comparison benchmark, also built an ensemble of models. We found that the TTA was a performance booster for the ensemble method and increased the final model's AUC score from 0.911 to 0.915, probably by reducing the overfitting in the augmented features.

In addition to the increased accuracy and generalizability of the results, the TTA technique can also potentially reduce the prediction bias that may result from inherent bias in the training dataset (i.e. if medical outcomes of specific subgroups did not result purely from clinical status) or from the model training (decreased accuracy for subgroups that are under-represented in the training set). In this work for example, the correlation between the prediction result and specific ethnicities was loosened to some extent by the TTA, as the final prediction resulted from averaging four prediction results, three that used the original ethnicity and one with a randomly assigned ethnicity group. The unbiasing of medical predictions from characteristics such as race and ethnicity is currently advocated as good practice [40].

The novelty of our work is the demonstration of the predictive power that results from combining massive ensemble methods with TTA techniques for tabular training data.

VIII. CODE AVAILABILITY

The preprocessing, stacking and test-time augmentation source code is available at https://github.com/sefficohen/ICU-Survival-Prediction.

ACKNOWLEDGMENT

The WiDS Datathon 2020 is a collaboration led by the Global WiDS team at Stanford, the West Big Data Innovation Hub, and the WiDS Datathon Committee. The authors thank the WiDS Datathon organizers and MIT GOSSIS initiative for the opportunity to analyze this data.

REFERENCES

[1] *WiDS Datathon 2021 Data*. Accessed: Jan. 6, 2021. [Online]. Available: https://www.kaggle.com/c/widsdatathon2021/data

[2] J. E. Zimmerman, A. A. Kramer, D. S. McNair, and F. M. Malila, "Acute physiology and chronic health evaluation (APACHE) IV: Hospital mortality assessment for today's critically ill patients," *Crit. Care Med.*, vol. 34, no. 5, pp. 1297–1310, May 2006.

[3] M. Lee, J. Raffa, M. Ghassemi, T. Pollard, S. Kalanidhi, O. Badawi, K. Matthys, and L. A. Celi, "WiDS (women in data science) datathon 2020: ICU mortality prediction (version 1.0.0)," PhysioNet, 2020, doi: 10.13026/vc0e-th79.

[4] J. Raffa, A. Johnson, L. A. Celi, T. Pollard, D. Pilcher, and O. Badawi, "33: The global open source severity of illness score (GOSSIS)," *Crit. Care Med.*, vol. 47, no. 1, p. 17, 2019.

[5] M. Michailidis, "Investigating machine learning methods in recommender systems," Ph.D. dissertation, Dept. Financial Comput., Univ. College London, London, U.K., 2017.

[6] A. Krizhevsky, I. Sutskever, and E. G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, 1097-1105.

[7] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019.

[8] F. Perez, C. Vasconcelos, S. Avila, and E. Valle, "Data augmentation for skin lesion analysis," in *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*. Cham, Switzerland: Springer, 2018, pp. 303–311.

[9] J. S. Friedland, J. C. Porter, S. Daryanani, J. M. Bland, N. J. Screaton, M. J. J. Vesely, G. E. Griffin, E. D. Bennett, and D. G. Remick, "Plasma proinflammatory cytokine concentrations, acute physiology and chronic health evaluation (APACHE) III scores and survival in patients in an intensive care unit," *Crit. Care Med.*, vol. 24, no. 11, pp. 1775–1781, Nov. 1996.

[10] J. R. Le Gall, "A new simplified acute physiology score (SAPS II) based on a European/North American multicenter study," *JAMA: J. Amer. Med. Assoc.*, vol. 270, no. 24, pp. 2957–2963, Dec. 1993.

[11] S. Lemeshow, "Mortality probability models (MPM II) based on an international cohort of intensive care unit patients," *JAMA: J. Amer. Med. Assoc.*, vol. 270, no. 20, pp. 2478–2486, Nov. 1993.

[12] S. F. B. Zaidi, M. A. Raouf, and T. Tariq, "Comparison of APACHE II, SAPS II and SOFA scoring systems as predictors of mortality in ICU patients," Pervaiz Ellahi Inst. Cardiol., Multan, Pakistan, Tech. Rep., 2019, vol. 53, doi: 10.7176/JMPB.

[13] T. A. Ayazoglu, "A comparison of APACHE II and APACHE IV scoring systems in predicting outcome in patients admitted with stroke to an intensive care unit," *Anaesthesia, Pain & Intensive Care*, vol. 15, no. 1, pp. 7–12, 2019.

[14] R. Lior, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 1–39, 2010.

[15] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997.

[16] *WiDS Datathon 2020 Data*. Accessed: Jan. 11, 2020. [Online]. Available: https://www.kaggle.com/c/widsdatathon2020/data

[17] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," in *Annals of Statistics*. Stanford, CA, USA: Stanford Univ., 2001, pp. 1189–1232.

[18] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[19] C.-Y. Lee and B.-S. Chen, "Mutually-exclusive-and-collectively-exhaustive feature selection scheme," *Appl. Soft Comput.*, vol. 68, pp. 961–971, Jul. 2018.

[20] *WiDS Datathon 2020 Leaderboard*. Accessed: Feb. 25, 2020. [Online]. Available: https://www.kaggle.com/c/widsdatathon2020/leaderboard

[21] M. Saar-Tsechansky and F. Provost, "Handling missing values when applying classification models," *J. Mach. Learn. Res.*, vol. 8, pp. 1623–1657, Jul. 2007.

[22] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6638–6648.

[23] *Surival Prediction Using Tensorflow 2.0*. Accessed: Jan. 27, 2020. [Online]. Available: https://medium.com/wids-mysore/surival-prediction-using-tensorflow-2-0-a547b3dc2112

[24] *Predicting Survival Using H_2O AutoML*. Accessed: Jan. 26, 2020. [Online]. Available: https://medium.com/wids-mysore/predicting-survival-using-h2o-automl-fcc1cf4605d6

[25] *2nd Place Solution*. Accessed: Feb. 25, 2020. [Online]. Available: https://www.kaggle.com/c/widsdatathon2020/discussion/132387

[26] *3rd Place Solution*. Accessed: Feb. 25, 2020. [Online]. Available: https://www.kaggle.com/c/widsdatathon2020/discussion/132292

[27] D. H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop challenges Represent. Learn. (ICML)*, vol. 3, Jun. 2013, p. 2.

[28] *4th Place Solution*. Accessed: Feb. 25, 2020. [Online]. Available: https://www.kaggle.com/c/widsdatathon2020/discussion/132312

[29] *5th Place Solution*. Accessed: Feb. 25, 2020. [Online]. Available: https://www.kaggle.com/c/widsdatathon2020/discussion/132267

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, and T. Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3146–3154.

[31] *6th Place Solution*. Accessed: Mar. 3, 2020. [Online]. Available: https://www.kaggle.com/c/widsdatathon2020/discussion/133509

[32] I. W. J. Guyon, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.

[33] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4765–4774.

[34] G. Fasano and A. Franceschini, "A multidimensional version of the Kolmogorov–Smirnov test," *Monthly Notices Roy. Astronomical Soc.*, vol. 225, no. 1, pp. 155–170, 1987.

[35] S. V. Buuren and K. Groothuis-Oudshoorn, "Mice: Multivariate imputation by chained equations in R," *J. Stat. Softw.*, vol. 45, no. 3, pp. 1–68, 2011.

[36] G. Chao, C. Mao, F. Wang, Y. Zhao, and Y. Luo, "Supervised nonnegative matrix factorization to predict ICU mortality risk," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2018, pp. 1189–1194.

[37] A. Sharma, A. Shukla, R. Tiwari, and A. Mishra, "Mortality prediction of ICU patients using machine leaning: A survey," in *Proc. Int. Conf. Compute Data Anal. (ICCDA)*, 2017, pp. 49–53.

[38] M. Aczon, D. Ledbetter, L. Ho, A. Gunny, A. Flynn, J. Williams, and R. Wetzel, "Dynamic mortality risk predictions in pediatric critical care using recurrent neural networks," 2017, *arXiv:1701.06675*. [Online]. Available: http://arxiv.org/abs/1701.06675

[39] T. Alves, A. Laender, A. Veloso, and N. Ziviani, "Dynamic prediction of ICU mortality risk using domain adaptation," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 1328–1336.

[40] D. A. Vyas, L. G. Eisenstein, and D. S. Jones, "Hidden in plain sight—Reconsidering the use of race correction in clinical algorithms," Harvard Med. School, Boston, MA, USA, Tech. Rep., 2020, pp. 874–882.

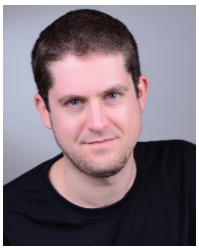
SEFFI COHEN received the B.Sc. degree in computer science from MTA, Tel-Aviv, in 2010, and the M.Sc. degree in computer science from Open University, Israel, in 2018. He is currently pursuing the Ph.D. degree in software and information systems engineering with Ben Gurion University.

He is also an AI Researcher. He founded the first Data Science Team, Israeli Defense Force, and led dozens of various AI solutions that were successfully deployed and yielded tangible operational value. He is a Kaggle master. He reached the top of the leader board in many international ML competitions.

NOA DAGAN received the M.D. and M.P.H. degrees from Hebrew University, and the Ph.D. degree in computer science from Ben-Gurion University. She is currently a Postdoctoral Fellow with the Department of Biomedical Informatics (DBMI), Harvard Medical School. She is the Director of data and AI-driven medicine with the Clalit Research Institute. She is a Public Health Physician and a Researcher. Her research interest focuses on practical implementations of machine learning algorithms using clinical data.

NURIT COHEN-INGER received the B.Sc. and M.Sc. degrees in computer science from Bar-Ilan University, in 1997 and 2018, respectively.

From 2017 to 2019, she was the Chief Data Officer with the Israeli Defense Forces. She is currently an AI Transformation Expert with 24 years of experience in leading and developing information systems. She is also the VP of Products at Beyondminds, an AI Software Provider. She led the IDF's AI transformation, planned AI strategy, built personnel skillset, and developed dozens of valuable AI solutions. Her main research interests include design and build applied AI solutions, building AI teams, machine learning systems architectures, and automatic machine learning platforms.



DAN OFER received the B.Sc. degree in psychobiology, in 2013, and the dual M.Sc. degree in bioinformatics and neurobiology from The Hebrew University of Jerusalem, Israel, in 2015.

He is currently a Senior Data Scientist with Medtronic, Israel. He is an AI Researcher. Previously, at SparkBeyond he developed analytics AI solutions in multiple industries, including insurance, finance, healthcare, and novel biomarker discovery with CRI. He researched AI methods for novel protein discovery with Prof. Michal Linial at the Hebrew University. His research interests include explainable AI, automated feature engineering on tabular data, proteomics, neuroplasticity, and AI in healthcare.

LIOR ROKACH is currently a Data Scientist and a Professor of software and information systems engineering (SISE) with the Ben-Gurion University of the Negev (BGU). He has established the Machine Learning Laboratory, BGU, which promotes innovative adaptations of machine learning and data science methods to create the next generation of intelligent systems. He is the author of over 300 peer-reviewed articles in leading journals and conference proceedings, patents, and book chapters. His research interests include design and analysis of machine learning and data mining algorithms and their applications in recommender systems, cyber security, and medical informatics.