EXTERNAL PROJECT: GENDER CLASSIFICATION OF FIRST NAMES USING LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORKS AND SUPPORT VECTOR MACHINE IN VARIOUS COUNTRIES

KRISTEL VAN ROOIJ

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

u1273632

COMMITTEE

dr. G.A. Chrupala
dr. P.H.G. Hendrix

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

May 21, 2021

ACKNOWLEDGMENTS

I would like to express my gratitude to dr. G.A. Chrupala for his support during the process of writing my master thesis. His constructive feedback and critical eye ensured that I could get the best out of myself. I would also like to thank Boudewijn van Straaten, my external supervisor at ASML, for his guidance and positive energy during this process. Lastly, I would like to thank dr. P.H.G. Hendrix for taking the time to read and review this study.

EXTERNAL PROJECT: GENDER CLASSIFICATION OF FIRST NAMES USING LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORKS AND SUPPORT VECTOR MACHINE IN VARIOUS COUNTRIES

KRISTEL VAN ROOIJ

Abstract

This study investigates to what extent multiple gender prediction models (i.e., Long Short-Term Memory (LSTM), Support Vector Machine (SVM) and Python gender-guesser baseline) accurately predict gender using first names and whether these models are sensitive to various countries. This study distinguishes itself from previous research since it compares different models for gender prediction, and it includes country as a feature to possibly improve performance of the models. The Gender Name Database is used to train, validate and test the models. An ASML dataset is used as additional test set to see how the models perform on different data. It was found that LSTM and SVM accurately predict gender using first names. The Python gender-guesser baseline is least suitable for gender prediction due to its high non-classification rate, either with or without country included. The performance of LSTM improves when country is included as a feature indicating that LSTM is sensitive to various countries. The performance of SVM with country feature is ambiguous depending on the dataset.

1 INTRODUCTION

This study examines to what extent multiple gender prediction models accurately predict gender using first names and whether these models are sensitive to various countries. The gender prediction models used are SVM and LSTM Recurrent Neural Networks (RNN). The Python package

gender-guesser is the baseline which is compared with the SVM and LSTM models to see if more advanced models are of added value. Additionally, and in contrast to previous research, this study examines whether the SVM and LSTM models perform differently in various countries. Country is included as one of the features to see if this can improve performance of the models.

This study is an external project at ASML. ASML is a Dutch high-tech company and leader in the semiconductor industry. This study is supportive of a larger Diversity & Inclusion (D&I) initiative to analyze how diverse and inclusive ASML currently is and where ASML can still improve. A part of this D&I initiative investigates possible (unconscious) gender bias in the recruitment funnel. The recruitment funnel refers to the process whereby candidates apply online, have a screening, a first interview, a second interview, and are hired or not hired. The most accurate model of this study will be used by ASML to analyze (unconscious) gender bias in their recruitment funnel. Gender of candidates who applied in the past two years will be predicted as this information is currently not available. Subsequently, the gender labels will be used to examine the dropout rate of female and male applicants in the recruitment funnel and whether there is gender bias.

Gender prediction of names is important from a societal, practical and scientific point of view. From a societal perspective, it is noted that there is a higher attention towards gender and gender equality whereas personal data shared is becoming less gender informative. It is often no longer mandatory to declare gender when you create an online account. Consequently, the gender of a substantial part of users is not known (Hu et al., 2021). Therefore, it is crucial to make automated gender predictions for marketing purposes (Antipov, Berrani, & Dugelay, 2016; Wais, 2016) such as targeted advertisements (Mueller & Stumme, 2016). Likewise, mobile and web applications benefit from gender prediction by the name of users (Lekamge & Fernando, 2019). Additionally, the need for gender classification by name can be seen in other areas as well, such as generating automated written communication with clients (e.g., use of pronouns) and performing demographic analyses (Bhagvati, 2018). Analyses related to gender are becoming more important to bring gender inequality to light and to improve women's inclusion. Algorithms that predict gender using other features, such as names, thereby give rise to improve data used in such analyses that did not contain gender information (Santamaría & Mihaljević, 2018).

Gender prediction of names is also important from a practical point of view. A dataset with all existing names and their genders does not exist. Predicting gender of names with machine learning solves the lack of

coverage of a dataset. Additionally, names continue to develop which also emphasizes the need of machine learning. Furthermore, gender prediction by name can be applied to coreference resolution (To, Van Nguyen, Nguyen, & Nguyen, 2020), which is a Natural Language Processing (NLP) task. Coreference resolution determines all textual references which point to the same entity (e.g., people) (Ferreira Cruz, Rocha, & Lopes Cardoso, 2020). Predicting gender by name can be applied to point textual references to the right entity. For example, mark 'He' to 'James' instead of incorrectly to 'Mia'. Coreference resolution is able to highly increase accuracy for other NLP tasks such as sentiment analysis (Sukthanker, Poria, Cambria, & Thirunavukarasu, 2020).

The research domain of gender prediction by name has been widely addressed. Researchers performed gender classification in numerous ways, for example using faces, Twitter messages, chats, handwriting and emails (Bhagvati, 2018). Recently, interest in gender classification of names has grown (Wais, 2016). Santamaría and Mihaljević (2018) compared various available web services for gender classification of names, such as NameAPI and genderize.io. Besides these available web services, researchers used various machine learning methods to predict gender by name. Examples are Deep Neural Network (DNN), Convolutional Neural Network (CNN) and LSTM (Bhagvati, 2018; Hu et al., 2021).

This study is scientifically relevant in multiple ways. First, there is a need to further examine performances of different models regarding gender classification of names (Santamaría & Mihaljević, 2018; Wais, 2016). This study is supportive to this need with a comparison of the Python gender-guesser baseline, LSTM and SVM. Second, the scope of this study is broader than previous studies. It examines whether the LSTM, SVM and baseline model perform differently in various countries. Including country as one of the features for the models to see if this improves gender prediction is not done in previous research. In summary, the comparison of different models and inclusion of country as a feature to possibly improve performance distinguishes this study from previous research. Section 2 elaborates on previous research.

This study answers the following research question:

RQ *To what extent do multiple gender prediction models accurately predict gender using first names and are these models sensitive to various countries?*

This research question is answered with two sub questions:

SQ1 *Are more advanced models (i.e., LSTM and SVM) of added value to predict gender of first names compared to the Python gender-guesser baseline?*

SQ2 *Does the performance of LSTM, SVM and the baseline model improve when country is included as a feature?*

This study found that LSTM and SVM models are of added value for gender prediction compared to the Python gender-guesser baseline. The Python gender-guesser is least suitable for gender prediction due to its high non-classification rate, either with or without country included. Performance of LSTM slightly improves when country is included as a feature, indicating that LSTM is sensitive to various countries. No one-sided conclusion can be drawn regarding SVM and its sensitively to various countries, since performance of SVM differs when country is or is not included depeding on the dataset used.

The remainder of this study is structured as follows. Section 2 elaborates on related work of gender prediction by name. Section 3 discusses the method and Section 4 explains the experimental setup. In Section 5, the results are presented. Section 6 provides a discussion and Section 7 concludes this study.

2 RELATED WORK

This section discusses related work of gender prediction of names. First, the area of research is explained. Subsequently, previous research related to available web services for gender prediction is discussed. Thereafter, various machine learning approaches used are discussed. Lastly, the goal and contribution of this study are explained.

2.1 *Area of research*

The area of research in this study is gender prediction of names. Recently, interest in this research area is increasing (Wais, 2016). Gender classification is performed in numerous ways. Besides names also faces, Twitter messages, handwriting and emails are used (Bhagvati, 2018). Names are very suitable for gender prediction since someone's name tells a lot about a person. Names disclose someone's gender, country of origin and ethnicity (Ye & Skiena, 2019).

Gender prediction of names is an issue of importance. Gender prediction of names is crucial for marketing purposes (Antipov et al., 2016; Wais, 2016), it can generate automated written communication with clients, it

can be used to perform demographic analyses (Bhagvati, 2018), it solves the lack of coverage of a dataset and it can be applied to coreference resolution (To et al., 2020). The inclusion of country to see if this improves performance of the models is important as the gender of a name can vary in different countries (Hu et al., 2021).

2.2 *Available web services for gender prediction by name*

This paragraph discusses previous studies using available web services for gender prediction by name. According to To et al. (2020), NameAPI and GenderAPI are the most commonly used web services for gender prediction. However, limitations of these services are that they are not open source and their dataset with names and gender is limited. Additionally, accuracy of these tools are low for some languages (e.g., Chinese).

Santamaría and Mihaljević (2018) argue that even though it is important to evaluate the error rate of gender predictions, there is limited research which compares various methods. In their own study, Santamaría and Mihaljević (2018) evaluated multiple available web services for gender prediction of names, specifically NameAPI, genderize.io, NamSor, Gender API and the Python gender-guesser. They found that, without parameter tuning, misclassification rate was lowest for the Python package gender-guesser which uses a table of names for look-up. However, due to its small dataset, the package showed poor performance looking at non-classifications (i.e., names for which gender could not be predicted due to absence in the dataset).

One of the few studies that compared different gender prediction methods is the research of Wais (2016). He evaluated gender prediction methods from highly regarded papers of Larivière, Ni, Gingras, Cronin, and Sugimoto (2013) and West, Jacquet, King, Correll, and Bergstrom (2013). Additionally, he compared these methods with the R package genderizeR which is an extension of the genderize.io service. Wais (2016) concluded that the genderize.io service is the best method since its non-classification rate was lowest. However, a limitation of the genderize.io is that it is mainly based on data which is retrieved from social media profiles from the United States (US) and England (Wais, 2016) and therefore may have a preference for English names (Santamaría & Mihaljević, 2018).

Karimi, Wagner, Lemmerich, Jadidi, and Strohmaier (2016) compared web services for gender prediction of names as well, specifically the Sexmachine and genderize.io. Karimi et al. (2016) argue that results of gender prediction using Sexmachine and genderize.io vary depending on the country of residence of someone.

2.3 *Machine learning methods for gender prediction by name*

In addition to available web services, previous studies also used machine learning to predict gender by name. Hu et al. (2021) investigated multiple character-based methods (e.g., DNN, CNN and LSTM) to classify gender using names retrieved from Yahoo! and American baby names. They argue that, using a linear model and DNNs, gender can be accurately predicted using information hidden in the spelling of first names.

Bhagvati (2018) performed gender classification of Indian, Sri Lankan, Japanese and Western names using CNN and LSTM. They investigated the influence of different word representations of first names (e.g., one-hot representation and word embeddings) on the performance of gender classification. Additionally, they introduced an enhanced integer representation of first names whereby padded zeros are replaced by random values between zero and one. Bhagvati (2018) concludes that LSTM is the best performing model using word embedding based on the introduced enhanced integer representation. The model achieves an accuracy of 84.30%. Similarly, Lekamge and Fernando (2019) conclude that LSTM is an appropriate model to predict gender of names. Using one-hot encoded Sri Lankan first names, the model achieves an accuracy of 94.94%. Likewise, To et al. (2020) found that the best gender prediction of names was achieved with LSTM. The LSTM outperformed models such as SVM, Logistic Regression and Decision Tree. It achieved an F1-score of 96% using fastText word embeddings of Vietnamese first and middle names.

A different method for gender prediction is used by Panchenko and Teterin (2014). They applied a linear supervised model together with three features (word endings, letter n-grams and a dictionary of names with a probability score that the name is masculine) to predict gender of 100,000 Russian full names retrieved from Facebook. The model reached an accuracy of 96%.

Previous research also used names from Twitter users for gender prediction. Wood-Doughty, Andrews, Marvin, and Dredze (2018) applied a RNN and CNN model using a dataset constructed by Burger, Henderson, Kim, and Zarrella (2011) consisting of almost 60,000 Twitter users. Additionally, SVM is used as a baseline. The SVM is based on previous work of Knowles, Carroll, and Dredze (2016) who used letter n-gram features derived from user names on Twitter. The research of Wood-Doughty et al. (2018) reports an accuracy score of 84.3%, 83.1% and 82.3% for the RNN, CNN and SVM respectively. They elucidate these accuracy scores with the fact that neural networks, like RNN and CNN, are able to learn more complex features compared to SVM.

2.4 *Current study*

The goal of this study is to examine to what extent multiple gender prediction models (i.e., LSTM, SVM and Python gender-guesser baseline) accurately predict gender using first names and whether these models are sensitive to various countries. Section 3 provides a motivation for the chosen models. As already briefly mentioned in Section 1, this study contributes to previous research in various ways.

First, Wais (2016) argues that there exists a need to report and compare performances of different gender classification models. Additionally, Santamaría and Mihaljević (2018) point out that most studies that perform gender classification of names do not motivate their chosen model. Furthermore, they state that only a handful of researchers compare different models. This study narrows this gap since it compares LSTM, SVM and the Python gender-guesser.

Second, in contradiction to previous research, this study is placed in a broader context since it examines whether the LSTM, SVM and baseline model perform differently in various countries. This study includes country as one of the features to see if this can improve performance of the models. Some previous studies predicted gender using names from multiple countries. However, these studies did not examine whether the model is sensitive to these countries. This is important since a name can be either feminine or masculine in different countries (Hu et al., 2021). For example, the name 'Aad' is considered masculine in the Netherlands and feminine in the United Kingdom (Hardvard Dataverse, 2018). In conclusion, comparing various models and including country as a feature to possibly increase performance distinguishes this study from previous research.

3 METHODS

This section explains the methods used. First, LSTM is explained. Subsequently, SVM is briefly discussed. Lastly, the Python gender-guesser baseline is explained. Please note that also the rationale for each model is described.

3.1 *LSTM*

The LSTM algorithm is a RNN type which is introduced by Hochreiter and Schmidhuber (1997). A RNN is called recurrent since the same task is executed for each item in the sequence (Bouktif, Fiaz, Ouni, & Serhani, 2018). A disadvantage of RNNs is that they suffer from the vanishing

gradient problem. To solve this problem, LSTM was designed which is able to learn long-term patterns via the use of gates (Bouktif et al., 2018).

A LSTM unit consists of three gates and a memory cell where information is stored. The input gate, forget gate and output gate manage the flow of information (C. J. Huang & Kuo, 2018). The gates learn to keep important sequence data and throw away unimportant sequence data. The weight, which is regulated by the LSTM, determines the importance of information (Q. Zhang, Gao, Liu, & Zheng, 2020). In this way, important information is passed through the network to make (gender) predictions.

Figure 1 (Variengien & Hinaut, 2020, p. 5) shows the LSTM structure which relies on the current input x_t and the output of the previous time step h_{t-1}. The gates include an element-wise operator X and a sigmoid σ which outputs a value between zero and one. A value of zero indicates that no information is let through and a value of one indicates that all information is let through. The forget gate decides how much information in the memory cell of the previous time step C_{t-1} should be forgotten or retained. The input gate determines which information should be stored in the memory cell of the current time step C_t. The output gate decides how much the output value of the current timestep h_t relies on the memory cell of the current time step C_t (Q. Zhang et al., 2020).

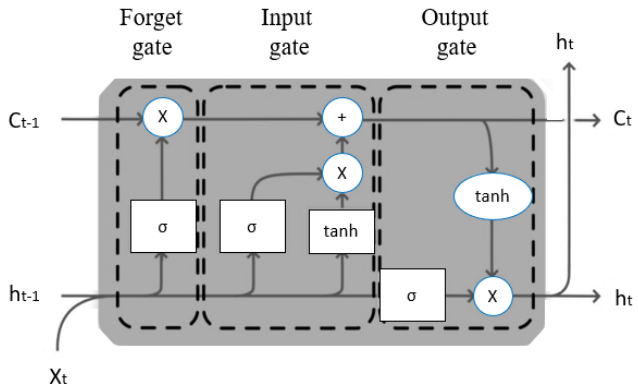

Figure 1: LSTM structure (Variengien & Hinaut, 2020, p. 5).

The input, forget and output gate are represented with Equation 1, 2 and 3 respectively (J. Zhang, Zhu, Zhang, Ye, & Yang, 2018).

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$

The equations include a sigmoid activation function σ. x_t is the input at time t and h_{t-1} is the activation vector of the hidden layer at the previous time step. Weight matrices are represented as W_i, W_f, W_o, U_i, U_f and U_o. Additionally, b_i, b_f and b_o are bias vectors. Equation 4, 5 and 6 show how the state of the memory cell C_t and hidden state h_t are updated (J. Zhang et al., 2018).

$$\tilde{C}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \tag{5}$$

$$h_t = o_t \otimes tanh(C_t) \tag{6}$$

Again, W_c and U_c are weight matrices and b_c is a bias vector. Equation 4 depends on the hidden layer at the previous time step h_{t-1} and the current input x_t. This equation uses a tanh activation function and outputs a vector of new possible values $\tilde{C}_t$. These values may be added to the current state of the memory cell C_t depending on the element-wise multiplication of the input gate i_t in the second part of Equation 5. The first part of Equation 5 consists of the element-wise multiplication of the forget gate f_t and the memory cell of the previous time step C_{t-1}. Lastly, Equation 6 shows that the hidden state is updated based on an element-wise multiplication of the output gate o_t and memory cell of the current time step C_t which is put through a tanh activation function.

LSTM is used since the model learns from sequence data, such as the sequence of letters in a name (Bhagvati, 2018). A character-based approach to classify gender is appropriate since particular characters, or a combination of characters, disclose the gender of an individual (Hu et al., 2021). Additionally, it is possible to include non-sequential data, such as country, in the LSTM as well. This is further explained in Section 4. Lastly and already mentioned, LSTM solves the vanishing gradient problem of RNNs and is therefore able to learn long-term dependencies.

3.2 *SVM*

The second method is the well known SVM, which is invented by Vapnik and Chervonenkis in 1963 (Gururaj, Shriya, & Ashwini, 2019). SVM separates two classes with a decision boundary based on similarities between features of every observation (Vogado, Veras, Araujo, Silva, & Aires, 2018). The decision boundary is also known as the hyperplane and is shown in Figure 2 (Yuan et al., 2017). The hyperplane is positioned such that it

maximizes the distance (i.e., margin) between the closest points of each class (i.e., support vectors) (S. Huang et al., 2018).

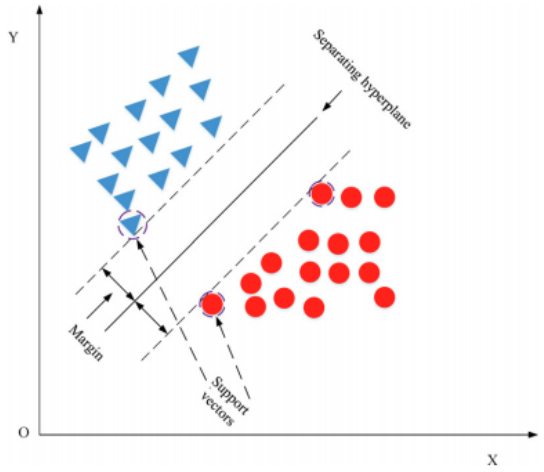

Figure 2: SVM visualization (Yuan et al., 2017, p. 57).

SVM is used since it is a powerful model to perform classification (S. Huang et al., 2018). Its performance is generally higher compared to other classification methods (Tripathi & Faruqui, 2011). Additionally, the number of studies using SVM to predict gender of names is quite limited. Therefore, there is a need to examine the performance of this model for gender classification.

3.3 *Python package gender-guesser*

The Python package gender-guesser is used as a baseline. It is a popular web service for gender prediction of names (Vasarhelyi & Vedres, 2021). The package is based on a dictionary of approximately 45,000 names and includes six possible outcomes (Santamaría & Mihaljević, 2018). Specifically, the gender prediction can be 'female', 'male', 'mostly_female', 'mostly_male', 'unknown' (when the name is not found in the dictionary) or 'andy' (when the probability for female and male is the same). Additionally, it is possible to specify the country of the name. This is one of the reasons why this package is used as a baseline. Furthermore, the package is used since the gender labels in the dictionary are of high quality as they are checked by native speakers of multiple countries (Santamaría & Mihaljević, 2018).

4 EXPERIMENTAL SETUP

This section describes the experimental setup. First, the raw datasets are described. Second, preprocessing of the datasets is described and sample creation is explained. Subsequently, transformation of the first name and country code features are described. Third, data is explored in more detail with an Exploratory Data Analysis (EDA). Fourth, the experimental procedure is discussed including an explanation how country is included in the models and how hyperparameters are tuned. Fifth, the software used is explained. Lastly, the evaluation metric is discussed.

4.1 *Raw datasets*

Three datasets are used. The first dataset is the Gender Name Database (GNDB) (Hardvard Dataverse, 2018). The GNDB consists of 6,277,039 observations and includes the features first name, gender, country code, gchar12, gchar1 and gchar2. Names from 182 different countries are included. The dataset is publicly available and downloaded as an Excel file from the Harvard Dataverse website. The original name of the dataset is wgnd_langctry.tab. The dataset is used to train, validate and test the models.

The second dataset is provided in an Excel format by ASML. The dataset is called ASML (ASML, 2021) and consists of 21,549 current ASML employees working in Europe and the US. Features which are included are first name, gender, nationality, employee number, prefix, last name, manpower group and region. This dataset is used as additional test set to see how the models perform on a different dataset.

The third dataset is retrieved from Datahub in a csv format and is called Country Codes (Datahub, 2019). Features which are included are country and a two letter country code. The dataset has 249 observations and the original name of the dataset is country-list_zip. This dataset is used to create a sample from the GNDB and ASML dataset, which is explained in the preprocessing paragraph of this section. Appendix A (page 32) provides a complete overview of the datasets with a definition of each variable and their descriptive statistics.

4.2 *Preprocessing data*

First, preprocessing of the GNDB is discussed and thereafter preprocessing of the ASML dataset is described. Additionally, it is explained how samples of these datasets are created. Lastly, the transformations of the features first name and country code are explained.

4.2.1 *Preprocessing GNDB*

Unnecessary variables in the GNDB were removed. Columns gchar12, gchar1 and gchar2 were empty and therefore deleted. Remaining features are first name, gender and country code. Next, rows with non-available values were dropped (i.e., 31 first names and five country codes). Additionally, 946 names consisted of Chinese characters. The main difference between Chinese and Latin languages is that Chinese is logo syllabic, and each letter has its own meaning (Jia & Zhao, 2019). The observations were dropped due to these differences and the fact that the ASML dataset does not contain Chinese characters. Subsequently, names that did not make sense (e.g., containing a lot of spaces) were deleted. Lastly, males were encoded with a zero and females with a one. Preprocessing resulted in a dataset with 6,275,922 observations.

4.2.2 *Preprocessing ASML dataset*

Likewise, unnecessary features in the ASML dataset were deleted. The features employee number, prefix, last name, manpower group and region were dropped. First name, gender and nationality were remaining features necessary to perform gender prediction. Non-available values were not found. First names containing special characters like dots were cleaned. Again, for the gender feature, a zero was used to represent a male and a one was used to represent a female. Preprocessing resulted in a dataset with 21,546 observations.

4.2.3 *Sample GNDB and ASML dataset*

Using all remaining observations from the GNDB was computationally too expensive. Therefore, a sample of the GNDB and ASML dataset was created. Based on the list available on (Worldometer, 2021), nineteen countries with the largest population worldwide were selected. Additionally, the Netherlands was selected as a country since the headquarters of ASML is located in the Netherlands and accordingly a large proportion of the employees have a Dutch nationality. The Country Codes dataset is merged with the twenty selected countries to provide these countries with a country code. An overview of selected countries with corresponding population and country code can be found in Appendix B (page 35). Additionally, nationalities in the ASML dataset are provided with a country code. Appendix C (page 36) provides a more detailed explanation of the sample creation and explains what is done about unisex labels which are not included in the raw datasets. Additionally, an overview of the samples and their descriptive statistics is provided.

To clarify, each row includes a first name, gender and country code. Furthermore, the GNDB sample includes country and the ASML sample includes nationality as a feature. For example, the name 'Aad' occurs four times in the GNDB sample. Specifically, three times with a feminine gender label (in the US, Nigeria and the Democratic Republic of Congo) and once with a masculine gender label (in the Netherlands).

4.2.4 *Transformation of features first name and country code*

First names, used as input for the LSTM, are represented using one-hot encoding. This means that "for the maximum word length L, and N distinct characters in dataset, a matrix V of size L × N is used, to represent a name" (Bhagvati, 2018, p. 617). The values of matrix V consist of ones (if character is present in name) and zeros (if character is not present in name). Names shorter than the maximum length are padded with an 'END' token. The initial maximum number of characters in a name was 33 and 27 for the GNDB and ASML sample respectively. A visual representation of the number of characters in first names can be found in Appendix D (page 39). Only 0.08% of the GNDB sample and 0.11% of the ASML sample consisted of names with more than twenty characters. Very sparse data may possibly hamper learning of LSTM (Bhagvati, 2018). Therefore, these names were dropped to make sure they would not make name matrices unnecessary large and possible hamper performance of the models. The resulting shape of a name is 20 by 28 (i.e., 26 letters of the alphabet, space token and 'END' token).

Likewise, the country code feature used as input for LSTM is one-hot encoded. Country code can be one-hot encoded based on character level or on the country code itself. This depends on the way country is included in LSTM. This is explained in the country feature paragraph of this section.

Following previous research (Mueller & Stumme, 2016; Wood-Doughty et al., 2018), character n-gram features of first names are used as input for SVM. Character n-gram features seek to identify groups of letters that often appear together in female and male names (Tripathi & Faruqui, 2011). Unigrams, bigrams and trigrams (i.e., 1-gram, 2-gram and 3-gram) are considered to represent first names. 4-gram features are not taken into account since they may result in overfitting on train data, and they are computationally more expensive (Tripathi & Faruqui, 2011).

Similarly, the country code feature used as input for SVM is represented with character n-gram features. Unigrams and bigrams are used since the maximum number of characters in a country code is two.

4.3 *Exploratory Data Analysis*

EDA is performed using Matplotlib (Hunter, 2007) to explore the samples (after dropping names with more than twenty characters) in more detail. Figure 3 provides a word cloud of first names in the GNDB sample. Word clouds visualize the most frequent words with specific colors and larger font size (Wang, Zhao, Guo, North, & Ramakrishnan, 2020). As shown in Figure 3, names such as Sara, Nadia and Marina are commonly used across countries, which aligns with the top five most occurring names provided in Appendix C (page 36). It makes sense that, for example, the name Sara appears in many countries since this is a well-known name from the Bible.

Figure 3: Word cloud of first names in GNDB sample.

Figure 4 and Figure 5 visualize the number of characters in first names. Figure 4 illustrates that most female names in the GNDB sample consist of seven characters and most male names include six characters. Looking at the ASML sample in Figure 5, most female names consist of six characters and most male names include five characters.

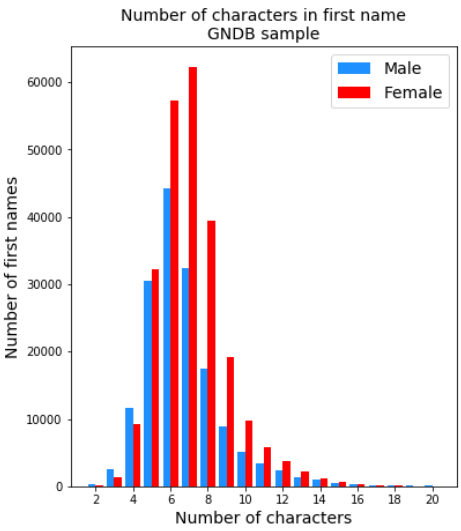

Figure 4: Length first names in GNDB sample.

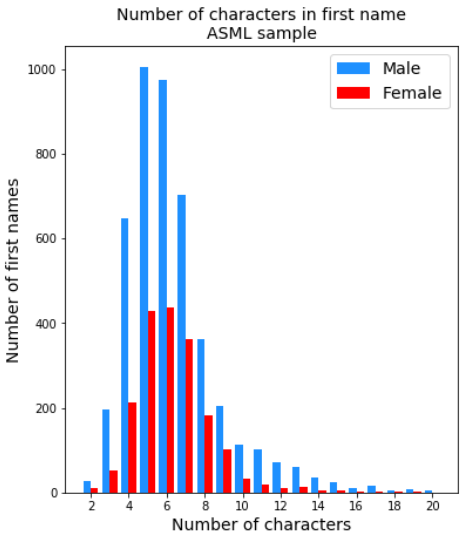

Figure 5: Length first names in ASML sample.

4.4 *Experimental procedure*

This paragraph explains the experimental procedure. First, it is explained how country code is included in the models. Second, hyperparameter tuning of LSTM and SVM is discussed.

4.4.1 *Including country feature*

Including country as one of the features for the models to see if this improves gender prediction is not done in previous research. Accordingly, the next paragraphs explain how country is included in the LSTM, SVM and Python gender-guesser baseline.

Two possibilities to include country in LSTM are used. The first possibility is to add the two-letter country code in front of the first name and use it as an input for LSTM. A dot is used as a character to separate the country code and name. For example, nl.aad indicates the Netherlands as a country and 'Aad' as a name. This string is used as an input for the LSTM. The resulting input shape is 23 by 29. The one-hot encoded matrix consists of 23 rows since the longest name consists of twenty characters, appended with a two letter country code and a dot as separator. The length of each row is 29 since there are 26 letters in the alphabet, a space token, an 'END' token and a dot token. The rest of the study refers to this option to include country as 'possibility one'.

The second possibility to include country in LSTM, is to concatenate the country code feature with the first name (of shape 20 by 28) after the first name went through a LSTM and dropout layer. The country code is represented as a one-hot encoding vector. The length of the vector is twenty, since twenty countries are included in the samples. The values of the vector consist of a one (if country code is present) and zeros (if country code is not present). For this second possibility, Keras functional API[1] is used. Keras functional API is able to design more flexible models compared to Keras sequential API[2] (which is used in possibility one). For example, Keras functional API is able to deal with multiple inputs (Chollet, 2015) such as names and countries. The rest of the study refers to this option to include country as 'possibility two'.

For the SVM without country, a pipeline from Scikit-learn (Pedregosa et al., 2011) is used. The pipeline takes in the names, performs a first transformation (i.e., creating n-grams) and subsequently performs a second transformation (i.e., the Support Vector Classifier (SVC)) to create a trained model which is used for gender prediction. However, since now multiple columns of the data frame (i.e., name and country code) are used, two pipelines are created. One pipeline for the name feature and one pipeline for the country code feature. First, the name and country pipeline apply a column-extracting transformer to select the relevant column (i.e., name or country code) and subsequently the pipelines perform a transformation to create n-grams. Thereafter, a third pipeline is created to combine the name and country pipelines. FeatureUnion from Scikit-learn (Pedregosa et al., 2011) is used to concatenate the outputs of the name and country pipelines before feeding it to the SVC.

Country is included in the Python gender-guesser baseline as well. Each of the twenty countries are represented in a suitable format for the Python gender-guesser. For example, the 'United States' was renamed as 'usa' to make sure the Python gender-guesser would recognize the country. The recognizable country names were retrieved from the Python gender-guesser documentation[3]. Countries that were not included in the Python gender-guesser package were renamed as 'other_countries'. A complete overview of the country codes, countries and renamed countries is provided in Appendix E (page 40).

[1] https://keras.io/guides/functional_api/
[2] https://keras.io/api/models/sequential/
[3] https://pypi.org/project/gender-guesser/

4.4.2 *LSTM hyperparameters*

Choosing optimal hyperparameters can make a distinction between an average and good performing LSTM (Reimers & Gurevych, 2017a). Hyperparameter optimization is performed using the validation set. This study considers a vanilla and stacked LSTM. A vanilla LSTM has a single hidden layer of LSTM units (Sagheer & Kotb, 2019). A stacked LSTM has multiple hidden layers of LSTM units (Bhagvati, 2018). In summary, six LSTM models are analyzed: a vanilla and stacked LSTM without country, a vanilla and stacked LSTM with country using possibility one, a vanilla and stacked LSTM with country using possibility two.

Following previous research (Lekamge & Fernando, 2019; Wood-Doughty et al., 2018), the LSTM models were trained using a binary cross entropy loss function and Adam optimizer. Additionally, a sigmoid activation function is used as gender prediction is considered a binary classification. Furthermore, experiments have been carried out with different batch sizes (i.e., 256, 512, 1024). In the end, a batch size of 1024 was chosen. The most optimal number of hidden nodes was determined using the Bayesian Optimization method. This method learns from training history and subsequently determines the next optimal possible hyperparameter setting (Reimers & Gurevych, 2017a). The minimum number of hidden nodes was set on 32 and the maximum was set on 128. The maximum number of trails to find the best setting of hidden nodes was set on three. Rerunning non-deterministic algorithms, such as LSTM, is important to make accurate conclusions (Reimers & Gurevych, 2017b). Therefore, each LSTM model was run three times with three different random seeds. As a result, the optimal number of hidden nodes, determined with the Bayesian Optimizer, may vary per run.

In addition to these hyperparameters, multiple regularization methods were applied to deal with overfitting. A dropout rate of 0.5 was used to make sure the LSTM did not become too sensitive to training data. Second, early stopping is used to end training before LSTM overfits (Baek & Kim, 2018). Early stopping was applied if the validation loss did not decrease for more than five epochs. Lastly, the best L2 regularization value (i.e., 0.01 or 0.1) has been chosen for each LSTM model based on accuracy averaged over three runs. The architectures and optimal hyperparameters settings of the LSTM models can be found in Appendix F (page 41).

4.4.3 *SVM hyperparameters*

The LinearSVC of Scikit-learn (Pedregosa et al., 2011) is used for SVM with and without country feature included. According to the Scikit-learn

documentation[4], LinearSVC is almost similar to a regular SVC with a linear kernel function. However, LinearSVC has more possibilities regarding loss functions and penalties. The default parameters of penalty (i.e., L2) and loss (i.e., squared hinge) have been used. LinearSVC has been chosen since it works well on sparse data (Yu et al., 2020) such as the n-grams features first name and country code. The regularization parameter C is tuned using validation data. After looping over various values of C (i.e., 0.01, 0.02, 0.04, 0.05, 0.10, 0.25), it was found that 0.04 was the most optimal hyperparameter setting for SVM with and without country included.

4.5 *Software*

Python 3.8.3 is used as programming language and the Jupyter Notebook (version 6.0.3) in Anaconda is used for processing. Keras (Chollet, 2015) with TensorFlow (Abadi et al., 2016) as backend was used to build the LSTM, Scikit-learn (Pedregosa et al., 2011) for the SVM and the Python package gender-guesser (Michael, 2008) was used to implement the baseline model. Additionally, other packages and libraries used are provided below.

- Pandas (McKinney, 2010)
- NumPy (Harris et al., 2020)
- Matplotlib (Hunter, 2007)
- Wordcloud (Halvey & Keane, 2007)

4.6 *Evaluation Metric*

Accuracy is used as evaluation metric for the models. Accuracy is calculated as the number of correctly classified instances (true negatives plus true positives) divided by the total number of instances (Chicco & Jurman, 2020). Accuracy is measured using gender labels in the GNDB sample, on which the models are trained, validated and tested. Additionally, gender labels in the ASML sample are used as additional test set to measure accuracy of the models on different data.

5 RESULTS

This section presents results of gender prediction by name using the LSTM, SVM and baseline model. First, performance of the models without country is provided. It is discussed whether more advanced models (i.e., LSTM

[4] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

and SVM) are of added value to predict gender of names compared to the Python gender-guesser baseline. Second, the performance of the three models with country included is presented. Lastly, a comparison is made between the models with and without country to see if performance of the models improves when country is included.

5.1 *Performance models without country included as a feature*

In this paragraph, results of the LSTM, SVM and Python gender-guesser baseline on the GNDB and ASML dataset are presented. Table 1 provides an overview of the (average) accuracy scores of the models. LSTM models were run three times with three different random seeds. Therefore, the accuracy is averaged over three runs. Additionally, standard deviations and minimum and maximum accuracy scores of the runs are provided.

Looking at the results of the Python gender-guesser baseline, something remarkable happens. Taking into account all predicted genders (i.e., either male, female or unknown), the baseline reports an accuracy of 14.98% on the GNDB and 55.63% on the ASML dataset. However, taking into account only male and female genders, the baseline reports an accuracy of 97.26% on the GNDB and an accuracy of 96.63% on ASML data. These results can be explained by the fact that the Python gender-guesser is based on a dictionary of approximately 45,000 names (Santamaría & Mihaljević, 2018). Since the GNDB test set consists of 24,389 unique first names and the ASML sample consist of 5,562 unique first names, the higher non-classification rate on the GNDB (i.e., 84.59% for GNDB and 42.43% for ASML) is not surprising. This result is in line with previous research from (Santamaría & Mihaljević, 2018) who argued that the misclassification rate was lowest for the Python gender-guesser but that performance was poor looking at non-classifications. One of the motivations for this study was that a dataset with all existing names and their genders does not exist and therefore machine learning should be applied to solve the lack of coverage of a dataset. From this point of view and the results of the baseline presented in Table 1, the baseline seems less suitable for gender prediction of names.

Regarding results of LSTM and SVM, the vanilla LSTM yields the best accuracy score on both the GNDB (86.22%) and ASML dataset (83.25%). These accuracy scores are shown in bold in Table 1. Results of the stacked LSTM are slightly worse, reporting an average accuracy of 86.19% on the GNDB and 83.17% on the ASML dataset. Predictions of LSTM models are considered stable, since the largest standard deviation is 0.52%. Performance of SVM is marginally lower than the stacked LSTM. Specifically, 85.31% on the GNDB and 83.12% on ASML data.

In conclusion, the baseline is least suitable for gender prediction of names and the vanilla LSTM is considered the best performing model for both the GNDB and ASML dataset. SVM performs slightly lower than LSTM. It can be concluded that more advanced models are of added value to predict gender of names compared to the Python gender-guesser baseline.

Table 1: (Average) accuracy scores LSTM, SVM and Python gender-guesser.

Model	**Test set**	**(Average) accuracy**	**Min**	**Max**	**St. dev.**
Vanilla LSTM	GNDB	**0.8622**	0.8614	0.8627	0.0007
	ASML	**0.8325**	0.8301	0.8339	0.0021
Stacked LSTM	GNDB	0.8619	0.8591	0.8645	0.0027
	ASML	0.8317	0.8284	0.8377	0.0052
SVM	GNDB	0.8531	-	-	-
	ASML	0.8312	-	-	-
Gender-guesser (with unknowns)	GNDB	0.1498	-	-	-
	ASML	0.5563	-	-	-
Gender-guesser (without unknowns)	GNDB	0.9726	-	-	-
	ASML	0.9663	-	-	-

To explore results of the best performing model in more detail, confusion matrices are presented in Table 2 and Table 3. As shown in Table 2, classification performance across females and males on the GNDB does not differ much. Specifically, 85.13% of males are correctly predicted and 87.04% of females are correctly predicted. Table 3 indicates that 85.10% of males in ASML data are correctly predicted and 79.09% of females are correctly predicted. Incorrectly predicted female names in the ASML dataset were further analyzed. It was found that 35.91% of Chinese female names (with a Latin alphabet) were predicted incorrectly as a male. It may be the case that Chinese female names contain more difficult and diverse naming conventions which make gender prediction more difficult.

Table 2: Confusion matrix Vanilla LSTM GNDB.

		Predicted	
		Male	Female
Actual	Male	20,880	3,648
	Female	4,760	31,961

Table 3: Confusion matrix Vanilla LSTM ASML.

		Predicted	
		Male	Female
Actual	Male	3,890	681
	Female	392	1,483

5.2 *Performance models with country included as a feature*

This paragraph presents results of LSTM, SVM and Python gender-guesser baseline with country included as a feature. Table 4 provides the (average) accuracy scores of the models. Idem to the baseline without country, the misclassification rate is low but results are poor looking at non-classifications.

Regarding classification performance on the GNDB, the vanilla LSTM with country included using possibility one yields the best average accuracy score of 86.58% (disregarding the baseline). The SVM performed slightly worse with an accuracy score of 85.31%.

Classification performance on the ASML dataset is somewhat different. This time, the stacked LSTM with country included using possibility two resulted in the best average accuracy of 83.92% (disregarding the baseline). Again, the SVM performed slightly worse with an accuracy score of 81.90%. The best accuracy scores are shown in bold in Table 4. Further, predictions of LSTM are considered stable, as the highest standard deviation is 0.52%.

Table 4: (Average) accuracy scores LSTM, SVM and Python gender-guesser with country.

Model	**Test set**	**(Average) accuracy**	**Min**	**Max**	**St. dev.**
Vanilla LSTM country possibility one	GNDB	**0.8658**	0.8653	0.8664	0.0005
	ASML	0.8384	0.8343	0.8415	0.0039
Stacked LSTM country possibility one	GNDB	0.8632	0.8621	0.8652	0.0017
	ASML	0.8368	0.8360	0.8382	0.0012
Vanilla LSTM country possibility two	GNDB	0.8632	0.8606	0.8646	0.0022
	ASML	0.8311	0.8255	0.8357	0.0052
Stacked LSTM country possibility two	GNDB	0.8651	0.8641	0.8656	0.0009
	ASML	**0.8392**	0.8356	0.8428	0.0036
SVM country	GNDB	0.8531	-	-	-
	ASML	0.8190	-	-	-
Gender-guesser country (with unknowns)	GNDB	0.0334	-	-	-
	ASML	0.3298	-	-	-
Gender-guesser country (without unknowns)	GNDB	0.9995	-	-	-
	ASML	0.9757	-	-	-

Table 5 and Table 6 present confusion matrices of the best performing models on the GNDB and ASML dataset. Table 5 indicates that accuracy scores across males and females are almost similar. Specially, the accuracy scores are 85.47% and 87.30% for the male and female class respectively.

Likewise, Table 6 presents quite similar accuracy scores for the male (84.53%) and female (81.17%) class.

Table 5: Confusion matrix Vanilla LSTM with country using possibility one on GNDB.

		Predicted	
		Male	Female
Actual	Male	20,963	3,565
	Female	4,664	32,057

Table 6: Confusion matrix Stacked LSTM with country using possibility two on ASML.

		Predicted	
		Male	Female
Actual	Male	3,864	707
	Female	353	1,522

5.3 *Comparison models with and without country included as a feature*

Now that results of the models with and without country are discussed, a comparison between them is presented in Table 7. The (average) accuracy scores of each model are shown. The best performing version per model (i.e., with or without country) for each dataset is shown in bold.

Performance of the vanilla LSTM is best when country is included using possibility one (86.58% and 83.84%). The performance of the stacked LSTM is highest when country is included using possibility two (86.51% and 83.92%). Classification performance of SVM on the GNDB slightly improves when country is included (85.31%). Note that this is not visible in Table 7 due to rounding. Contrary, performance on ASML data is higher for SVM without country (83.12%). The accuracy score for the baseline considering females, males and unknowns is higher without country (14.98% and 55.63%). However, performance of the baseline considering only males and females is slightly higher when country is included (99.95% and 97.57%).

Disregarding the baseline, the Vanilla LSTM with country included using possibility one yields the best performance on the GNDB with an average accuracy score of 86.58%. Regarding the ASML dataset, the stacked LSTM with country using possibility two performs best with an average accuracy score of 83.92%. These accuracy scores are shown in bold and underlined in Table 7.

Looking at the small accuracy differences between the GNDB and ASML dataset, it can be concluded that the models, which are trained and validated on the GNDB, are generalizable to the ASML dataset as well. Additionally, an overview of the training and validation accuracy plots of LSTM models is presented in Appendix G (page 43).

In conclusion, performance of LSTM slightly improves when country is included as a feature. Performance of SVM differs when country is or is not included depending on the dataset used. The baseline model is still

considered least suitable for gender prediction of names due to its poor results looking at non-classifications.

Table 7: (Average) accuracy scores LSTM, SVM and Python gender-guesser.

Model	**Test set**	**(Average) accuracy without country**	**(Average) accuracy country possibility one**	**(Average) accuracy country possibility two**
Vanilla LSTM	GNDB	0.8622	**0.8658**	0.8632
	ASML	0.8325	**0.8384**	0.8311
Stacked LSTM	GNDB	0.8619	0.8632	**0.8651**
	ASML	0.8317	0.8368	**0.8392**
SVM	GNDB	0.8531	**0.8531**	-
	ASML	**0.8312**	0.8190	-
Gender-guesser (with unknowns)	GNDB	**0.1498**	0.0334	-
	ASML	**0.5563**	0.3298	-
Gender-guesser (without unknowns)	GNDB	0.9726	**0.9995**	-
	ASML	0.9663	**0.9757**	-

6 DISCUSSION

This section provides a discussion. First, the goal of this study is provided. Second, the ethical aspect is considered. Third, findings are presented and put into perspective with links to existing literature. Fourth, limitations and suggestions for future research are discussed. Lastly, contribution of this study is explained.

6.1 *Goal of the study*

The goal of this study is to examine to what extent multiple gender prediction models accurately predict gender using first names and whether these models are sensitive to various countries. Gender prediction models used are LSTM and SVM. Furthermore, the Python gender-guesser is used as a baseline. The GNDB is used to train, test and validate the models. Additionally, the ASML dataset is used as an additional test set to see how models perform on different data. To achieve the goal, this study first examines whether more advanced models (i.e., LSTM and SVM) are of added value to predict gender of names compared to the Python gender-guesser

baseline. Second, this study investigates if performance of LSTM, SVM and the baseline model improve when country is included as a feature.

6.2 *Ethics*

This study is supportive of a D&I initiative at ASML. Predicting gender of ASML applicants, which are currently unknown, is essential to investigate possible (unconscious) bias and improvement opportunities related to gender diversity in the recruitment funnel. Despite the fact that this initiative is not part of this study, the ethical aspect should be considered due to the inclusion of first names from ASML applicants.

The purpose for the initiative is clear, being a diverse company is key to enhance equality between females/males, and to attract and retain talent. Additionally, research has shown that a diverse workforce increases company performance (Cho, Kim, & Mor Barak, 2017). However, despite this purpose, it should be recognized that some people may object to have their gender automatically inferred without their consents. Nonetheless, there are two arguments why this study and initiative can be ethically justified.

First, only first names of applicants are used to perform gender classification. Last names are not included. This enhances privacy of applicants since there are numerous people with the same first name.

Second, data will be pseudonymized (GDPR Art. 4(5), 2016) after gender classification of first names is completed. First names are not used to investigate the dropout rate of female and male applicants in the recruitment funnel, only gender labels are used. After classifying candidates as female/male, first names are deleted and can only be assessed in the original dataset which is kept separately and under supervision of the respective department.

6.3 *Findings*

First, this study examines whether more advanced models (i.e., LSTM and SVM) are of added value to predict gender of names compared to the Python gender-guesser baseline. Unless its low misclassification rate, it was found that the Python gender-guesser baseline is least suitable for gender prediction due to its high non-classification rate of 84.59% on the GNDB and 42.43% on the ASML dataset. This result is in line with previous research of Santamaría and Mihaljević (2018) who also found a poor performance of the gender-guesser looking at non-classifications. Regarding results of more advanced models, it was found that the vanilla LSTM yield the best accuracy score of 86.22% on the GNDB and 83.25%

on the ASML dataset. A result which aligns with gender prediction of Indian, Sri Lankan, Japanese and Western names performed by Bhagvati (2018) who reported an accuracy score of 84.30% using LSTM. The accuracy score of 94.94% achieved by Lekamge and Fernando (2019) is higher. They argue that a high accuracy was reached since their dataset was restricted to Sri Lankan first names. They suggest expanding their dataset with first names from various countries in order to train LSTM for more difficult and diverse naming conventions. Accuracy scores of this study achieved by SVM were slightly lower than LSTM (i.e., 85.31% on the GNDB and 83.12% on ASML data). This finding aligns with the study of To et al. (2020) who conclude that gender prediction of LSTM slightly outperformed SVM. In conclusion, it was found that more advanced models are of added value to predict gender of names compared to the Python gender-guesser baseline.

Second, this study investigates if performance of LSTM, SVM and the baseline model improve when country is included as a feature. It was found that the performance of LSTM slightly improves on both datasets when country is included. Performance of SVM on the GNDB slightly improves as well. Contrary, the accuracy score of SVM on ASML data is higher when country is not included. Therefore, regarding SVM, no one-sided conclusion can be drawn. Even with country included, the baseline is still considered least suitable for gender prediction by name due to its poor results looking at non-classifications. The overview of all models with and without country included, presented in Table 7, shows that the vanilla LSTM with country included using possibility one yields the best accuracy score of 86.58% on the GNDB and the stacked LSTM with country included using possibility two yields the best accuracy score of 83.92% on the ASML dataset.

6.4 *Limitations and future research*

This study has limitations which give rise for future research. First, unisex gender labels are not considered. Gender prediction is considered as a binary classification. Binary gender prediction of names is in line with previous research (Hu et al., 2021; Jia & Zhao, 2019; To et al., 2020; Wood-Doughty et al., 2018). Furthermore, Wais (2016) argues that predefined unisex gender labels, such as in a dataset, are not very informative as they do not say anything about the probability that the name is feminine or masculine. The raw GNDB and ASML dataset only contained female and male gender labels. However, to narrow the limitation, unisex names used in the sampled countries, retrieved from Mom Junction (2021) and Family Education (2021), were dropped. Including unisex gender labels for gender prediction may be a direction for future research.

Second, as mentioned, no one-sided conclusion can be drawn whether SVM improves when country is included since the performance differed depending on the dataset used. This indicates that more research is needed to examine whether the performance of SVM improves when country is included.

Third, only Latin first names are included. Names with Chinese characters were not considered due to the differences between Chinese and Latin languages (e.g., Chinese is logo syllabic, and each character has its own meaning). Future research could investigate whether findings of this study also hold for the Chinese language.

6.5 *Contribution of the study*

This study contributes to the research area in several ways. First, there exist a need to further investigate performance of various models regarding gender classification of names (Santamaría & Mihaljević, 2018; Wais, 2016). This study supports this need with a comparison of the LSTM, SVM and Python gender-guesser baseline. Second, contrary to previous research, this study is placed in a broader context since it examines whether the LSTM, SVM and baseline model perform differently in various countries. This is important since names can be either masculine or feminine in various countries (Hu et al., 2021). Third, this study is supportive for diversity initiatives such as the D&I initiative at ASML. Models created in this study can support decision making for business regarding people management for topics such as D&I.

7 CONCLUSION

This study examines to what extent multiple gender prediction models accurately predict gender using first names and to whether these models are sensitive to various countries. Two sub questions are used to answer the research question.

The first sub question is *'Are more advanced models (i.e., LSTM and SVM) of added value to predict gender of first names compared to the Python gender-guesser baseline?'* It was found that LSTM and SVM models are of added value to predict gender compared to the Python gender-guesser baseline, which is least suitable for gender prediction due to its high non-classification rate.

The second sub question is *'Does the performance of LSTM, SVM and the baseline model improve when country is included as a feature?'* It was found that the performance of LSTM slightly improves when country is included. Regarding SVM, no one-sided conclusion can be drawn since performance

of SVM differs when country is or is not included depending on the dataset. Even with country included, the baseline model is still considered least suitable for gender prediction of names due to its high non-classification rate.

The answers on the sub questions jointly answer the research question *'To what extent do multiple gender prediction models accurately predict gender using first names and are these models sensitive to various countries?'* The Python gender-guesser baseline is least suitable for gender prediction, either with or without country included. LSTM and SVM models both accurately predict gender using first names. The performance of LSTM improves when country is included, indicating that LSTM is sensitive to various countries. No one-sided conclusion can be drawn regarding SVM and its sensitivity to various countries.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... others (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Antipov, G., Berrani, S.-A., & Dugelay, J.-L. (2016). Minimalistic CNN-based ensemble model for gender prediction from face images. *Pattern Recognition Letters*, *70*, 59–65.

ASML. (2021). ASML Employees [Data file].

Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications*, *113*, 457–480.

Bhagvati, C. (2018). Word representations for gender classification using deep learning. *Procedia Computer Science*, *132*, 614–622.

Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2018). Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies*, *11*(7), 1–20.

Burger, J. D., Henderson, J., Kim, G., & Zarrella, G. (2011). *Discriminating gender on twitter* (Tech. Rep.). MITRE Corporation.

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews Correlation Coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 1–13.

Cho, S., Kim, A., & Mor Barak, M. E. (2017). Does diversity matter? Exploring workforce diversity, diversity management, and organizational performance in social enterprises. *Asian Social Work and Policy Review*, *11*(3), 193–204.

Chollet, F. (2015). Keras. Retrieved from https://keras.io/

Datahub. (2019). country-list_zip [Data file]. Retrieved from https://datahub.io/core/country-list

Family Education. (2021). The ultimate guide to Mexican names [List with unisex Mexican names]. Retrieved from https://www.familyeducation.com/the-ultimate-guide-to-mexican-names

Ferreira Cruz, A., Rocha, G., & Lopes Cardoso, H. (2020). Coreference resolution: Toward end-to-end and cross-lingual systems. *Information*, *11*(2), 1-23.

GDPR Art. 4(5). (2016). General Data Protection Regulation. Retrieved from https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

General Authority for Statistics. (2017). National Code of Countries and Nationalities [List with nationalities and their country code].

Retrieved from https://www.stats.gov.sa/sites/default/files/national_directory_of_states_and_nationalities2017en.pdf

Gururaj, V., Shriya, V., & Ashwini, K. (2019). Stock market prediction using linear regression and support vector machines. *International Journal of Applied Engineering Research*, *14*(8), 1931–1934.

Halvey, M. J., & Keane, M. T. (2007). An assessment of tag presentation techniques. In *Proceedings of the 16th International Conference on World Wide Web* (pp. 1313–1314).

Hardvard Dataverse. (2018). wgnd_langctry [Data file]. Retrieved from https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/YPRQH8

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... others (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hu, Y., Hu, C., Tran, T., Kasturi, T., Joseph, E., & Gillingham, M. (2021). What's in a name? Gender classification of names with character based machine learning models. *Data Mining and Knowledge Discovery*, 1–27.

Huang, C. J., & Kuo, P. H. (2018). A deep CNN-LSTM model for particulate matter ($PM_{2.5}$) forecasting in smart cities. *Sensors*, *18*(7), 1-22.

Huang, S., Cai, N., Pacheco, P. P., Narrandes, S., Wang, Y., & Xu, W. (2018). Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genomics-Proteomics*, *15*(1), 41–51.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *IEEE Annals of the History of Computing*, *9*(03), 90–95.

Jia, J., & Zhao, Q. (2019). Gender prediction based on Chinese name. In *CCF International Conference on Natural Language Processing and Chinese Computing* (pp. 676–683).

Karimi, F., Wagner, C., Lemmerich, F., Jadidi, M., & Strohmaier, M. (2016). Inferring gender from names on the web: A comparative evaluation of gender detection methods. In *Proceedings of the 25th International Conference Companion on World Wide Web* (pp. 53–54).

Knowles, R., Carroll, J., & Dredze, M. (2016). Demographer: Extremely simple name demographics. In *Proceedings of the First Workshop on NLP and Computational Social Science* (pp. 108–113).

Larivière, V., Ni, C., Gingras, Y., Cronin, B., & Sugimoto, C. R. (2013). Bibliometrics: Global gender disparities in science. *Nature News*, *504*(7479), 211–213.

Lekamge, T., & Fernando, T. (2019). Finding the gender of personal names and finding the effect of Gana on personal names with Long Short

Term Memory. In *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)* (Vol. 250, pp. 1–8).

McKinney, W. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

Michael, J. (2008). Dictionary of first names and gender. Retrieved from https://pypi.org/project/gender-guesser/

Mom Junction. (2021). Unisex baby names with weaning [List with names and their gender]. Retrieved from https://www.momjunction.com/baby-names/unisex/page/37/

Mueller, J., & Stumme, G. (2016). Gender inference using statistical name characteristics in twitter. In *Proceedings of the The 3rd Multidisciplinary International Social Networks Conference on SocialInformatics 2016, Data Science 2016* (pp. 1–8).

Panchenko, A., & Teterin, A. (2014). Detecting gender by full name: Experiments with the Russian language. In *International Conference on Analysis of Images, Social Networks and Texts* (pp. 169–182).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Reimers, N., & Gurevych, I. (2017a). Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*.

Reimers, N., & Gurevych, I. (2017b). Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. *arXiv preprint arXiv:1707.09861*.

Sagheer, A., & Kotb, M. (2019). Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing*, *323*, 203–213.

Santamaría, L., & Mihaljević, H. (2018). Comparison and benchmark of name-to-gender inference services. *PeerJ Computer Science*, *4*, 1–29.

Sukthanker, R., Poria, S., Cambria, E., & Thirunavukarasu, R. (2020). Anaphora and coreference resolution: A review. *Information Fusion*, *59*, 139–162.

To, H. Q., Van Nguyen, K., Nguyen, N. L.-T., & Nguyen, A. G.-T. (2020). Gender prediction based on Vietnamese names with machine learning techniques. *arXiv preprint arXiv:2010.10852*.

Tripathi, A., & Faruqui, M. (2011). Gender prediction of Indian names. In *Ieee technology students' symposium* (pp. 137–141).

Variengien, A., & Hinaut, X. (2020). A journey in ESN and LSTM Visualisations on a language task. *arXiv preprint arXiv:2012.01748*.

Vasarhelyi, O., & Vedres, B. (2021). Gender typicality of behavior predicts

success on creative platforms. *arXiv preprint arXiv:2103.01093*.

Vogado, L. H., Veras, R. M., Araujo, F. H., Silva, R. R., & Aires, K. R. (2018). Leukemia diagnosis in blood slides using transfer learning in CNNs and SVM for classification. *Engineering Applications of Artificial Intelligence*, *72*, 415–422.

Wais, K. (2016). Gender prediction methods based on first names with genderizeR. *The R Journal*, *8*(1), 17–37.

Wang, J., Zhao, J., Guo, S., North, C., & Ramakrishnan, N. (2020). Recloud: Semantics-based word cloud visualization of user reviews. In *Graphics interface 2014* (pp. 151–158). AK Peters/CRC Press.

West, J. D., Jacquet, J., King, M. M., Correll, S. J., & Bergstrom, C. T. (2013). The role of gender in scholarly authorship. *PloS One*, *8*(7), 1-6.

Wood-Doughty, Z., Andrews, N., Marvin, R., & Dredze, M. (2018). Predicting Twitter user demographics from names alone. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media* (pp. 105–111).

Worldometer. (2021). Countries in the world by population [List with countries and their population]. Retrieved from https://www.worldometers.info/world-population/population-by-country/

Ye, J., & Skiena, S. (2019). The secret lives of names? Name embeddings from social media. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 3000–3008).

Yu, L., Wang, C., Chang, H., Shen, S., Hou, F., & Li, Y. (2020). Application research of intelligent classification technology in enterprise data classification and gradation system. *Complexity*, *2020*, 1–9.

Yuan, Y., Zhang, M., Luo, P., Ghassemlooy, Z., Lang, L., Wang, D., ... Han, D. (2017). SVM-based detection in visible light communications. *Optik*, *151*, 55–64.

Zhang, J., Zhu, Y., Zhang, X., Ye, M., & Yang, J. (2018). Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. *Journal of Hydrology*, *561*, 918–929.

Zhang, Q., Gao, T., Liu, X., & Zheng, Y. (2020). Public environment emotion prediction model using LSTM network. *Sustainability*, *12*(4), 1-116.

APPENDIX A: DESCRIPTIVE STATISTICS RAW DATASETS

Table 8: Features raw GNDB.

Feature	**Column name**	**Description**
Gender	gender	A person's gender. A male is encoded as 0 and a female is encoded as 1. Target of this paper
First name	name	A person's first name
Country code	code	Two letter country code with the first name in use
gchar12	N/A	Empty column and therefore deleted
gchar1	N/A	Empty column and therefore deleted
ghar2	N/A	Empty column and therefore deleted

Table 9: Descriptive statistics raw GNDB (6,277,039 observations).

Feature	**Unique observations**	**Percentage female/male**	**Top five most occurring names and country codes**
Gender	2	60.60% female 39.40% male	-
First name	174,820	-	Marina Maria Sara Olga Jana
Country code	182	-	CM CA KN KE US

Table 10: Features raw ASML dataset.

Feature	**Column name**	**Description**
Gender	Gender	A person's gender. A male is encoded as 0 and a female is encoded as 1. Target of this paper
First name	First Name	A person's first name
Nationality	Nationality	A person's nationality
Region	Region	The region where a person is working (US/Europe)
Prefix	Prefix	The prefix of a person's name
Last name	Last Name	A person's last name
Manpower group	Manpower Group	Whether a person is a flex or fix employee (flex/fix)
Employee number	Empl. Nr.	A person's employee number

Table 11: Descriptive statistics raw ASML dataset (21,549 observations).

Feature	**Unique observations**	**Percentages binary features**	**Top five most occurring names and nationalities**
Gender	2	17.95% female 82.05% male	-
First name	6,983	-	Peter Jeroen Micheal Paul Mark
Nationality	116	-	Dutch American Indian Chinese Belgian
Region	2	26.62% US 73.38% Europe	-
Prefix	35	-	-
Last name	13,617	-	-
Manpower group	2	7.09% flex 92.91% fix	-
Employee number	21,549	-	-

Table 12: Features raw Country Codes dataset (249 observations).

Feature	**Column name**	**Description**
Country	Name	Country
Country code	Code	Two letter country code

APPENDIX B: OVERVIEW COUNTRIES WITH THEIR RANK, COUNTRY CODE AND POPULATION

Table 13: Selected countries for the GNDB and ASML sample. Countries are selected based on the list named 'Countries in the world by population' available on (Worldometer, 2021). The Philippines (rank 13) is not selected since the Philippines is not included in the ASML dataset. Additionally, the Netherlands (rank 69) is selected since the headquarters of ASML is located in the Netherlands and accordingly a large proportion of the employees have a Dutch nationality.

Rank of largest countries in the world by population	**Country**	**Country code**	**Population**
1	China	CN	1,439,323,776
2	India	IN	1,380,004,385
3	United States	US	331,002,651
4	Indonesia	ID	273,523,615
5	Pakistan	PK	220,892,340
6	Brazil	BR	212,559,417
7	Nigeria	NG	206,139,589
8	Bangladesh	BD	164,689,383
9	Russia	RU	145,934,462
10	Mexico	MX	128,932,753
11	Japan	JP	126,476,461
12	Ethiopia	ET	114,963,588
14	Egypt	EG	102,334,404
15	Vietnam	VN	97,338,579
16	DR Congo	CD	89,561,403
17	Turkey	TR	84,339,067
18	Iran	IR	83,992,949
19	Germany	DE	83,783,942
20	Thailand	TH	69,799,978
69	Netherlands	NL	17,134,872

APPENDIX C: EXPLANATION SAMPLE CREATION AND DESCRIPTIVE STATISTICS

After sampling based on the twenty countries, the sample of the GNDB contained 408,341 observations with 162,850 unique first names. Regarding the ASML sample, one extra preprocessing step was necessary. Nationalities in the ASML sample had to be provided with a country code. This was done using the list with nationalities and country codes in the National Code of Countries and Nationalities documentation (General Authority for Statistics, 2017). Additionally, identical observations (based on first name, gender and country code) were dropped. This resulted in an ASML sample of 6,676 observations with 5,628 unique first names.

One constraint of the datasets which should be recognized is that they only include binary gender labels (male/female). Unisex gender labels are not included. To narrow this limitation, unisex names used in the twenty countries were dropped. These unisex names were retrieved from Mom Junction (2021) and Family Education (2021). Additionally, rows with an identical name and identical country code, but a different gender label were dropped in the ASML sample, since these observations indicate a unisex gender label as well. This preprocessing step resulted in a GNDB sample with 408,290 observations and 162,829 unique first names and a ASML sample with 6,453 observations and 5,562 unique first names. Additionally, predicted gender labels of the Python gender-guesser were renamed. In order to measure accuracy with binary gender labels of the GNDB and ASML sample, the label 'mostly_male' was renamed as 'male' and 'mostly_female' was renamed as 'female'. Furthermore, 'andy' and 'unknown' were both considered as unknowns. Section 6 discusses the rationale to consider gender prediction as a binary classification.

As a last preprocessing step, the GNDB sample was split in a train, validation and test set. The sets were stratified by first name regardless of the country code. This means that all observations with, for example, the name 'Aad' are included in either the train, validation or test set. 70% of the GNDB sample is reserved for training, 15% for model validation and 15% to test the models. A random state was used to create a reproducible split for the different models. There was no need to split the ASML sample since this sample is complete used as additional test set. An overview of the samples and their descriptives is provided on the next page.

Table 14: Features GNDB sample.

Feature	**Column name**	**Description**
Gender	gender	A person's gender. A male is encoded as 0 and a female is encoded as 1. Target of this paper
First name	name	A person's first name
Country code	code	Two letter country code with the first name in use
Country	Country	Country with the first name in use

Table 15: Descriptive statistics GNDB sample (408,290 observations).

Feature	**Unique observations**	**Percentage female/male**	**Top five most occurring**
Gender	2	60.11% female 39.89% male	-
First name	164,829	-	Sara Nadia Tania Marina Rita
Country code	20	-	NG US CD DE NL
Country	20	-	Nigeria United States DR Congo Germany Netherlands

Table 16: Features ASML sample.

Feature	**Column name**	**Description**
Gender	gender	A person's gender. A male is encoded as 0 and a female is encoded as 1. Target of this paper
First name	name	A person's first name
Country code	code	Two letter country code with the first name in use
Nationality	Nationality	A person's nationality

Table 17: Descriptive statistics ASML sample (6,453 observations).

Feature	**Unique observations**	**Percentage female/male**	**Top five most occurring**
Gender	2	29.06% female 70.94% male	-
First name	5,562	-	Ali Juan Gabriel Daniel Omar
Country code	20	-	NL US IN CN TR
Nationality	20	-	Dutch American Indian Chinese Turkish

APPENDIX D: NUMBER OF CHARACTERS IN FIRST NAMES

Figure 6 shows that most female names in the GNDB sample include seven characters and most male names consist of six characters. Figure 7 of the ASML sample shows that most female names include six characters and most male consist of five characters. The maximum number of characters is lower for females (i.e., 29 in GNDB and 19 in ASML) than for males (i.e., 33 in GNDB and 27 in ASML).

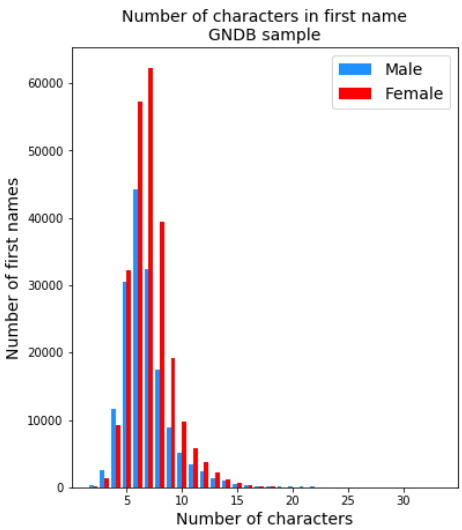

Figure 6: Length first names in GNDB sample.

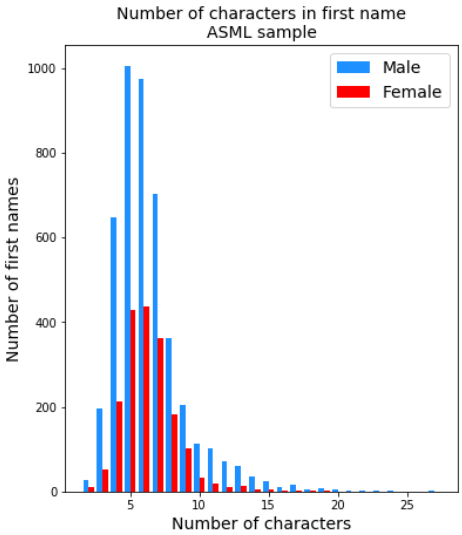

Figure 7: Length first names in ASML sample.

APPENDIX E: OVERVIEW OF COUNTRY CODE, COUNTRY AND RENAMED COUNTRY FOR PYTHON GENDER-GUESSER

Table 18: Overview of selected countries with their country code and renamed country for the Python gender-guesser baseline. The nationality column (present in the ASML sample) is added as well, since this column is used to create the 'renamed country' in the ASML sample. The renamed countries were retrieved from the Python gender-guesser documentation[5]. Countries that were not included in the Python gender-guesser package were renamed as 'other_countries'.

Country code	**Country**	**Nationality**	**Renamed country**
CN	China	Chinese	china
IN	India	Indian	india
US	United States	American	usa
ID	Indonesia	Indonesian	other_countries
PK	Pakistan	Pakistani	other_countries
BR	Brazil	Brazilian	other_countries
NG	Nigeria	Nigerian	other_countries
BD	Bangladesh	Bangladeshi	other_countries
RU	Russia	Russian	russia
MX	Mexico	Mexican	other_countries
JP	Japan	Japanese	japan
ET	Ethiopia	Ethiopian	other_countries
EG	Egypt	Egyptian	other_countries
VN	Vietnam	Vietnamese	vietnam
CD	DR Congo	Congolese	other_countries
TR	Turkey	Turkish	turkey
IR	Iran	Iranian	other_countries
DE	Germany	German	germany
TH	Thailand	Thai	other_countries
NL	Netherlands	Dutch	the_netherlands

[5] https://pypi.org/project/gender-guesser/

APPENDIX F: ARCHITECTURES LSTM

Table 19: Architecture first run of Vanilla LSTM without country feature using a batch size of 1024, dropout 0.5, L2 0.01, sigmoid activation, binary cross entropy loss function and Adam optimizer.

Layer	Output shape	Parameters
LSTM	(None, 96)	48,000
Dropout	(None, 96)	0
Dense	(None, 1)	97
Activation	(None, 1)	0

Table 20: Architecture first run of Stacked LSTM without country feature using a batch size of 1024, dropout 0.5, L2 0.10, sigmoid activation, binary cross entropy loss function and Adam optimizer.

Layer	Output shape	Parameters
LSTM	(None, 20, 128)	80,384
Dropout	(None, 20, 128)	0
LSTM	(None, 128)	131,584
Dropout	(None, 128)	0
Dense	(None, 1)	129
Activation	(None, 1)	0

Table 21: Architecture first run of Vanilla LSTM with country feature (possibility one) using a batch size of 1024, dropout 0.5, L2 0.10, sigmoid activation, binary cross entropy loss function and Adam optimizer.

Layer	Output shape	Parameters
LSTM	(None, 96)	48,384
Dropout	(None, 96)	0
Dense	(None, 1)	97
Activation	(None, 1)	0

Table 22: Architecture first run of Stacked LSTM with country feature (possibility one) using a batch size of 1024, dropout 0.5, L2 0.10, sigmoid activation, binary cross entropy loss function and Adam optimizer.

Layer	**Output shape**	**Parameters**
LSTM	(None, 23, 96)	48,384
Dropout	(None, 23, 96)	0
LSTM	(None, 96)	74,112
Dropout	(None, 96)	0
Dense	(None, 1)	97
Activation	(None, 1)	0

Table 23: Architecture first run of Vanilla LSTM with country feature (possibility two) using a batch size of 1024, dropout 0.5, L2 0.01, sigmoid activation, binary cross entropy loss function and Adam optimizer.

Layer	**Output shape**	**Parameters**
Input (name)	[(None, 20, 28)]	0
LSTM	(None, 96)	48,000
Dropout	(None, 96)	0
Input (country code)	[(None, 20)]	0
Concatenate	(None, 116)	0
Dense	(None, 1)	117
Activation	(None, 1)	0

Table 24: Architecture first run of stacked LSTM with country feature (possibility two) using a batch size of 1024, dropout 0.5, L2 0.10, sigmoid activation, binary cross entropy loss function and Adam optimizer.

Layer	**Output shape**	**Parameters**
Input (name)	[(None, 20, 28])	0
LSTM	(None, 20, 128)	80,348
Dropout	(None, 20, 128)	0
LSTM	(None, 128)	131,584
Dropout	(None, 128)	0
Input (country code)	[(None, 20)]	0
Concatenate	(None, 148)	0
Dense	(None, 1)	149
Activation	(None, 1)	0

APPENDIX G: TRAINING AND VALIDATION PLOTS LSTM (FIRST RUN)

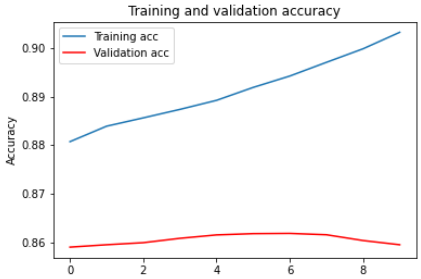

Figure 8: Vanilla LSTM.

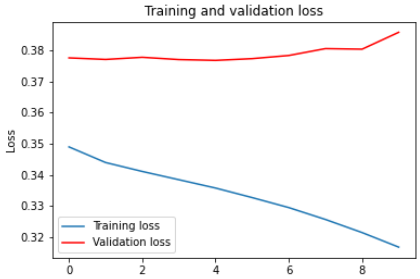

Figure 9: Vanilla LSTM.

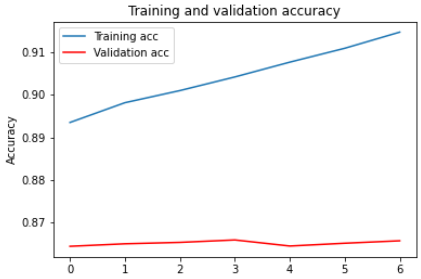

Figure 10: Stacked LSTM.

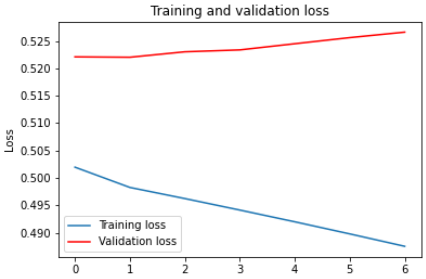

Figure 11: Stacked LSTM.

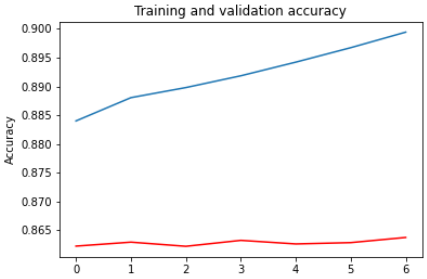

Figure 12: Vanilla LSTM country using possibility one.

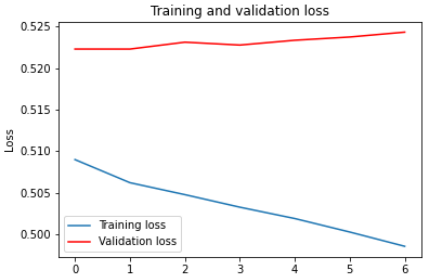

Figure 13: Vanilla LSTM country using possibility one.

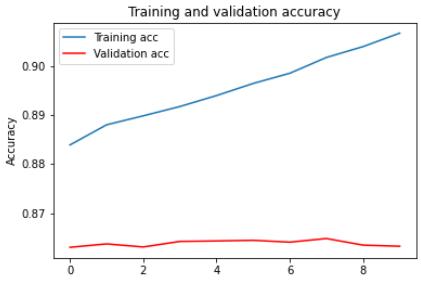

Figure 14: Stacked LSTM country using possibility one.

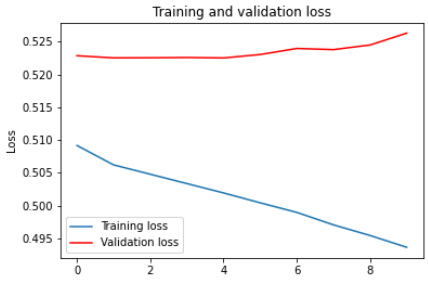

Figure 15: Stacked LSTM country using possibility one.

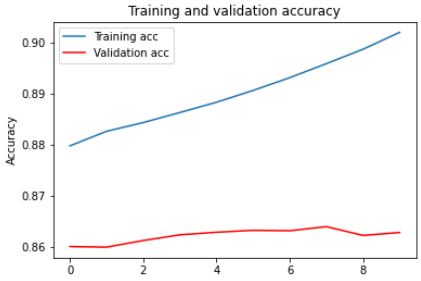

Figure 16: Vanilla LSTM country using possibility two.

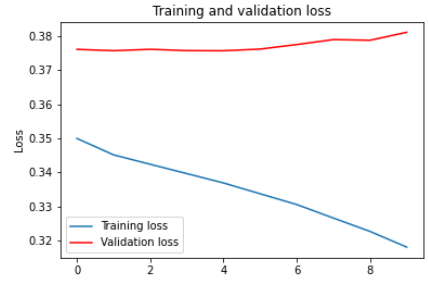

Figure 17: Vanilla LSTM country using possibility two.

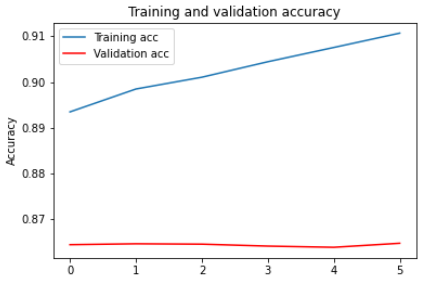

Figure 18: Stacked LSTM country using possibility two.

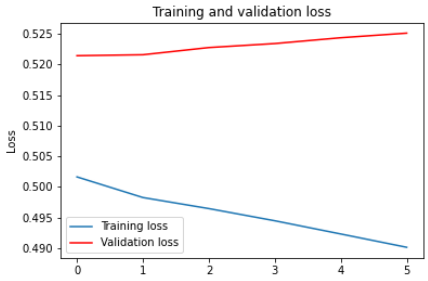

Figure 19: Stacked LSTM country using possibility two.