HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 2 (2023)

The Age of the Meta-Doctor: Diagnosing Parkinson's Disease with Artificial Intelligence and Speech

Ayush Tripathi

Cumberland Valley High School

ABSTRACT

The basal ganglia consist of the striatum, substantia nigra, and other nuclei, forming various pathways of motor initiation. Parkinson's disease (PD) is a neurodegenerative disorder characterized by dysfunction of the basal ganglia pathways. Consequently, PD affects the production of speech. An AI model can analyze audio samples from regular and PD patients. A simple deep learning model with various layers, ReLU activation, sigmoid activation, optimizer, loss function, and Early_Stopping can use extracted speech features to classify patients as regular or PD-afflicted with up to 97% accuracy. Overall, the advent of user-friendly artificial intelligence has led to exciting times, with new medical advancements emerging day after day; perhaps the ease of AI implementation will encourage others to solve everyday problems with just a computer and a dream.

Introduction

For much of history, humans have led a subpar existence. Disease, poor hygiene, and injury have taken the lives of millions. However, modern medicine has taken considerable strides in preserving lives in the past century. Sharing knowledge through the Internet has enabled researchers to communicate findings with academics on the opposite side of the world. This has led to an exponential increase in collective intelligence, where anyone with an Internet connection can research and solve problems. However, this revolution is not only in the medical field; in the last 100 years, the utilization of computers and robotic devices to automate repetitive tasks and optimize inefficient processes has had a widespread effect on the economy. Jobs based on these repetitive tasks have almost been phased out in developed regions, but occupations requiring more critical thinking are yet to be replaced. Medical professionals make complex diagnoses regularly, with the lives of their patients on the line. Thus, in the past decade, while the Internet and automation have greatly benefited the medical field, they have yet to become practical enough to make diagnoses with the regularity and complexity that a licensed medical professional can. That is, until the past few years.

In these few years, artificial intelligence has taken off and demonstrated the ability to achieve competence in complex tasks. This means computers can now perform critical thinking tasks, including medical diagnosis. However, implementing these technologies in the medical field takes much work. Still, it can be accelerated if medical professionals currently working can be taught to utilize artificial intelligence technologies for diagnosis. Unfortunately, the field of artificial intelligence is daunting for a beginner not knowing where to start. This paper aims to show that integrating medicine with AI is more straightforward than it initially seems. Combining biology and computer science will truly bring about the rise of the meta-doctor. Such an individual can diagnose, treat, and cure patients' diseases more accurately and precisely.

It is necessary to remember the motivation behind the information that will be presented. Not every facet of the topic will be explored; the goal is to gain enough knowledge so an effective model for Parkinson's

disease (PD) can be developed. Shortcuts and simplifications may be taken to maintain clarity and concision, so a certain degree of expertise in neuroscience and programming is assumed. The paper will first describe the function and dysfunction of the basal ganglia. Then, the methods for creating an AI-based PD diagnostic tool will be discussed.

Basal Ganglia

The brain's basal ganglia are deep, gray matter structures primarily involved in motor initiation and control (Ellens & Leventhal, 2014; Lanciego et al., 2012). The basal ganglia can be subdivided into input, intrinsic, and output nuclei (Lanciego et al., 2012). The striatum is the primary input nucleus, though newer studies have shown that other parts of the brain can also serve as input nuclei (Ellens & Leventhal, 2014). The intrinsic nuclei consist of the globus pallidus externus and the subthalamic nucleus, and the extrinsic nuclei are the globus pallidus internus and the substantia nigra reticulata (Lanciego et al., 2012). The substantia nigra compacta is a modulator that controls the neurological flow of information between specific nuclei and others, a concept that will be expanded upon throughout the paper (Ellens & Leventhal, 2014; Lanciego et al., 2012).

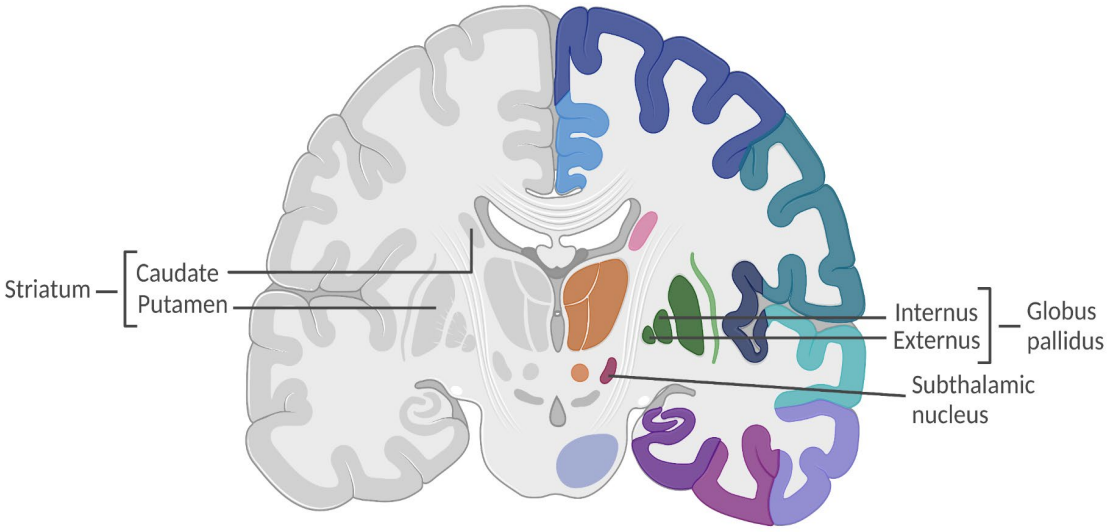


Figure 1. Basal ganglia of the brain. Created using BioRender.com. Created and copyrighted by Ayush Tripathi.

Striatum

The striatum (caudate and putamen) is a basal ganglia structure that contains interneurons and striatofugal neurons (Lanciego et al., 2012). Four main types of interneurons control the activity of striatofugal neurons (Lanciego et al., 2012). First, cholinergic interneurons involve the neurotransmitter acetylcholine and are called tonically-active neurons (always active, called TAN) (Lanciego et al., 2012). There is a decrease in the activity of TANs called the TAN pause that corresponds to a decrease in motor activity controlled by dopamine (Ellens & Leventhal, 2014). Other interneurons incorporate parvalbumin (GABAergic fast-spiking interneurons or FSI), calretinin, or nitric oxide (nitrergic interneurons) (Lanciego et al., 2012). These calretinin and nitrergic interneurons connect with TAN and FSI to make up striatal circuits essential for motor function and cognition (Ellens & Leventhal, 2014; Lanciego et al., 2012).

Striatofugal neurons leave the striatum and relay information to other nuclei in the brain (Lanciego et al., 2012). Striatofugal neurons are called medium-sized spiny neurons (MSNs) because the dendrites of these neurons have large branches with processes that give the appearance of spiny spikes (Kreitzer & Malenka, 2008). Most neurons in the striatum are MSNs (Lanciego et al., 2012). These neurons receive many inputs from the cortex and serve as "filters" of neurological information (Lanciego et al., 2012). The MSNs release GABA, an inhibitory transmitter, when activated. (Lanciego et al., 2012; Yin, 2017).

Striosomes and Matrix

The striatum has different neurons but no apparent dichotomy in cytoarchitecture (Lanciego et al., 2012). However, it has two histochemically dichotomous regions (Lanciego et al., 2012). Dendrites and axon collaterals of MSNs originate from either the matrix (darker when the striatum is dyed) or the striosome (Ellens & Leventhal, 2014). Efferent MSNs from striosomes (lighter when the striatum is dyed) typically project to a structure called the SNc directly, with offshoots to other nuclei (Lanciego et al., 2012). On the other hand, MSNs from the matrix typically project to structures called the SNr, GPe, and GPi (Lanciego et al., 2012). The SNc, SNr, GPe, and GPi will be discussed in depth later in the text, but it is crucial to understand how neurons from just one structure—the striatum—can branch out and project to many varying deep brain nuclei. The dichotomy between striosomes and matrix is not only for efferent neurons; afferent neurons also go to either striosome or matrix, depending on their origin (Lanciego et al., 2012).

Striatal Afferent Neurons

The cortex has various connections to the striatum, and for this paper, they will be referred to as corticostriatal connections (Ellens & Leventhal, 2014). Most relevant to the basal ganglia and Parkinson's disease are pyramidal layer V neurons, which innervate striosomes (Lanciego et al., 2012). There are two types of pyramidal layer V neurons: pyramidal tract neurons and intratelencephalic neurons (Ellens & Leventhal, 2014). Pyramidal tract neurons (PT-type) are offshoots of corticospinal neurons that ipsilaterally synapse to striatal MSNs as part of the indirect pathway (Lanciego et al., 2012). Intratelencephalic—within the telencephalon--neurons are another type (IT-type) and mainly synapse with striatal MSNs of the direct pathway (Lanciego et al., 2012).

Glutamatergic thalamostriatal neurons are another type of afferent neuron (Kreitzer & Malenka, 2008). Every interneuron and projection neuron is connected to thalamostriatal neurons (Kreitzer & Malenka, 2008). In addition, thalamostriatal neurons connect to the dendritic shafts of other neurons (Lanciego et al., 2012). These neurons originate from the midline thalamic nuclei, intralaminar thalamic nuclei, and ventral thalamic motor nuclei (Lanciego et al., 2012). The difference between the thalamostriatal and corticostriatal systems is that both have different glutamate transporters (Lanciego et al., 2012). The former generally contains the vGlut 2 glutamate transporter, while the latter generally has the vGlut 1 glutamate transporter-–though it has been noted that there is some overlap (Lanciego et al., 2012).

Striatal Efferents and Other Basal Ganglia

There are three main efferent pathways from the striatum: the two striatopallidal pathways (striatum-GPe and striatum-GPi) and the striatonigral pathway (striatum-SNr) (Lanciego et al., 2012). MSNs projecting to the SNr and GPi are modulated by D1R dopaminergic receptors, while MSNs projecting to the GPe are modulated by D2R receptors (Lanciego et al., 2012). Because it may be unclear, the exact form and function of the SNr, GPi, and GPe will be further explained later in the text, but regardless of the specifics, the core concept is that activated MSNs inhibit the activity of their relevant nuclei (Lanciego et al., 2012; Yin, 2017). D1R-containing

MSNs and D2R-containing MSNs do not activate simultaneously; for example, if D1R-containing MSNs are activated, then D2R-containing MSNs are generally inhibited (Ellens & Leventhal, 2014; Lanciego et al., 2012). This mechanism will be further discussed in the context of the direct and indirect pathways later in the text.

Globus Pallidus

The globus pallidus is a deep brain structure medial to the putamen and inferior to the caudate (Lanciego et al., 2012). The globus pallidus contains two parts: the globus pallidus internus (the GPi) and the globus pallidus externus (GPe) (Lanciego et al., 2012). The GPi is the inner part of the globus pallidus, and the GPe is the outer part of the globus pallidus (Lanciego et al., 2012). As mentioned earlier, the globus pallidus internus is modulated by GABAergic input from D1R-containing MSNs, while the globus pallidus externus is modulated by GABAergic input from D2R-containing MSNs (Kreitzer & Malenka, 2008; Lanciego et al., 2012). In addition, the GPe forms a direct information loop with the subthalamic nucleus (STN), another basal ganglia component (Ellens & Leventhal, 2014).

Subthalamic Nucleus

The subthalamic nucleus indeed synapses with GABAergic neurons from the GPe, but the STN can also receive direct information from the cortex in what is known as the hyperdirect pathway (Ellens & Leventhal, 2014; Lanciego et al., 2012; Yin, 2017). The hyperdirect pathway does not go through various other nuclei; instead, there is a direct connection between the subthalamic nucleus and the motor cortices and frontal eye nuclei (Yin, 2017). This shorter pathway requires less time (Lanciego et al., 2012; Yin, 2017). The excitatory glutamatergic STN efferents innervate the GPi and SNr (Lanciego et al., 2012). The subthalamic nucleus also sends and receives information to and from various other nuclei, which will not be discussed (Lanciego et al., 2012).

Substantia Nigra

Another critical nucleus, the substantia nigra, is the site of many dopaminergic neurons (Ellens & Leventhal, 2014). The substantia nigra is characterized by a dark color due to the compound neuromelanin (Lanciego et al., 2012). These neuromelanin-containing dopaminergic neurons primarily extend to the striatum, and degeneration of these neurons leads to Parkinson's disease (Yin, 2019). In PD, the remaining neurons are characterized by many misfolded proteins like alpha-synuclein that collect to form Lewy bodies (Chen et al., 2019; Wu & Hallett, 2013). The presence of PD symptoms is a strong suggestion that the substantia nigra pars compacta (SNc), the switch in the classical basal ganglia model, is not functioning correctly (Lanciego et al., 2012). Continual, unregulated degeneration of dopaminergic neurons in the substantia nigra can cause total impairment of motor function (Ellens & Leventhal, 2014; Kreitzer & Malenka, 2008; Lanciego et al., 2012; Wu & Hallett, 2013). When functioning normally, the SNc stimulates D1R receptors and inhibits D2R receptors (Lanciego et al., 2012). Another component of the substantia nigra is the substantia nigra pars reticulata (SNr). The SNr, along with the GPi, is an extrinsic nucleus involved in the direct and indirect pathways (Ellens & Leventhal, 2014; Kreitzer & Malenka, 2008). Both pathways are responsible for the initiation or inhibition of the movement and will be discussed shortly (Ellens & Leventhal, 2014; Lanciego et al., 2012).

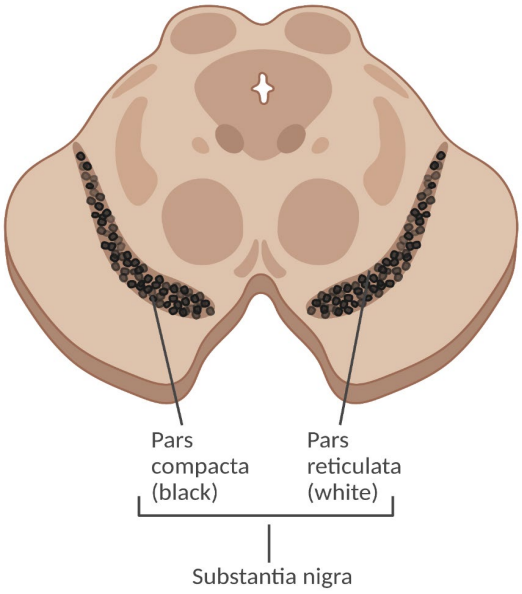


Figure 2. A cross-section of the midbrain showing the substantia nigra. Created using BioRender.com. Created and copyrighted by Ayush Tripathi.

The Direct and Indirect Pathways and SNc

Striatal MSNs involved with the direct pathway are called dMSNs (Ellens & Leventhal, 2014). The direct pathway is a circuit that involves GABAergic striatal MSNs inhibiting the substantia nigra pars reticulata and globus pallidus internus (Ellens & Leventhal, 2014; Kreitzer & Malenka, 2008; Lanciego et al., 2012). At the resting state, the SNr and GPi would inhibit the thalamus, a double-ovoid structure at the center of the cerebrum (Ellens & Leventhal, 2014; Kreitzer & Malenka, 2008; Lanciego et al., 2012). As a result, the thalamus would not activate the motor cortices of the cerebral cortex, and no movement occurs (Ellens & Leventhal, 2014; Lanciego et al., 2012). However, when the dMSNs inhibit SNr and GPi, the SNr and GPi cannot inhibit the thalamus (Ellens & Leventhal, 2014; Kreitzer & Malenka, 2008; Lanciego et al., 2012). The thalamus can then activate the motor cortices of the cerebral cortex, and movement occurs (Ellens & Leventhal, 2014; Lanciego et al., 2012).

Striatal MSNs involved with the indirect pathway are called iMSNs (Ellens & Leventhal, 2014). The indirect pathway is a circuit that involves iMSNs, the GPe, the STN, GPi, and SNr (Ellens & Leventhal, 2014; Kreitzer & Malenka, 2008; Lanciego et al., 2012). At the resting state, GPe sends inhibitory GABAergic signals to the STN (Ellens & Leventhal, 2014; Lanciego et al., 2012). However, when the iMSNs send GABAergic input to GPe, GPe neuronal activity is inhibited (Ellens & Leventhal, 2014; Kreitzer & Malenka, 2008; Lanciego et al., 2012). As a result, the inhibited GPe cannot inhibit the STN (Ellens & Leventhal, 2014; Lanciego et al., 2012). The STN will then send excitatory signals to the GPi and the SNr (Ellens & Leventhal, 2014; Lanciego et al., 2012). The GPi and SNr inhibit the neurons of the thalamus, and the thalamus cannot activate the motor cortices of the cerebral cortex (Ellens & Leventhal, 2014; Lanciego et al., 2012). Consequently, no movement occurs (Ellens & Leventhal, 2014; Kreitzer & Malenka, 2008; Lanciego et al., 2012). In the direct pathway, the striatum directly communicates with the GPi and SNr. However, in the indirect pathway, the striatum indirectly communicates with the GPi and SNr through other basal ganglia.

The SNc modulates between the direct and indirect pathways (Ellens & Leventhal, 2014; Lanciego et al., 2012). Striatal dMSNs contain D1R receptors, while striatal iMSNs contain D2R receptors (Lanciego et al.,

2012). When the SNc releases dopamine to the striatum, the dopamine binds to the D1R receptors, exciting the dMSNs (Ellens & Leventhal, 2014; Lanciego et al., 2012). Then, the GABAergic dMSNs inhibit the activity of the SNr and the GPi (Ellens & Leventhal, 2014; Lanciego et al., 2012). Since the SNr and GPi inhibit thalamic activity at the resting phase, inhibiting the SNr and GPi allows the thalamus to stimulate the motor cortices, initiating movement (Ellens & Leventhal, 2014; Lanciego et al., 2012). In short, the direct pathway is all about inhibiting the inhibitor. On the other hand, dopamine binding to D2R receptors produces an inhibitory effect rather than an excitatory effect (Ellens & Leventhal, 2014; Lanciego et al., 2012). As a result, GABAergic iMSN activity is inhibited (Ellens & Leventhal, 2014; Lanciego et al., 2012). Since inhibitory GABAergic iMSNs are themselves inhibited, the iMSN efferent, GPe, becomes active and sends inhibitory GABAergic signals to the STN (Ellens & Leventhal, 2014; Lanciego et al., 2012). The STN generally sends excitatory glutamatergic signals to the SNr and the GPi, but these excitatory signals are not sent if the GPe inhibits the STN (Ellens & Leventhal, 2014; Lanciego et al., 2012). Therefore, the indirect pathway will be halted (Ellens & Leventhal, 2014; Lanciego et al., 2012). However, if the SNc did not release dopamine, the direct pathway would not be activated (Ellens & Leventhal, 2014; Lanciego et al., 2012). Instead, in the indirect pathway, STN would be activated (Ellens & Leventhal, 2014; Lanciego et al., 2012). The STN would then activate the inhibitory SNr and GPi (Ellens & Leventhal, 2014; Lanciego et al., 2012). The SNr and GPi would inhibit the activity of the thalamus, and the indirect pathway would be complete; movement would not occur (Ellens & Leventhal, 2014; Lanciego et al., 2012). Considering the importance of dopamine and the SNc, it is unsurprising that the malfunction of dopaminergic input from the SNc is associated with PD (Ellens & Leventhal, 2014; Lanciego et al., 2012). The SNc is critical to initiating movement (Ellens & Leventhal, 2014; Lanciego et al., 2012). The classical basal ganglia model describes the SNc as a switch between the indirect and direct pathways (Lanciego et al., 2012). The classical model does not consider other connectivities between the basal ganglia (Lanciego et al., 2012). Lanciego et al. (2012) indicate that newer models that build upon the classical model have been proposed. However, the classical model is enough to provide an intuitive understanding of movement initiation for this paper.

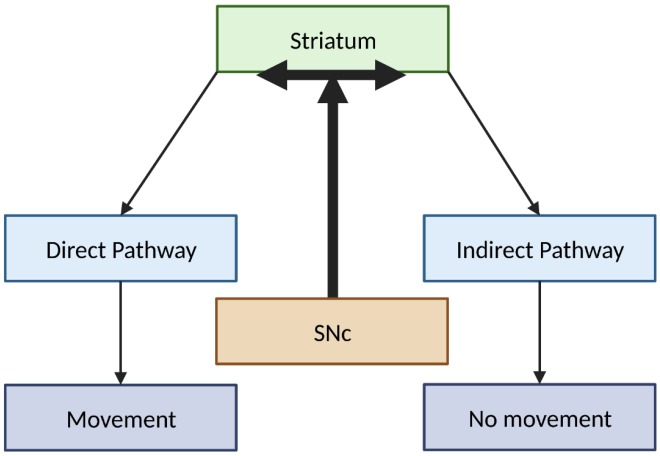


Figure 3. The SNc switches between the direct and indirect pathways in the classical basal ganglia model. Created using BioRender.com. Created and copyrighted by Ayush Tripathi.

Cerebellum

The cerebellum is a structure responsible for coordinating muscle activity (Wu & Hallett, 2013). While the cerebellum is not a part of the basal ganglia, it is a part of the metencephalon and is located beneath the occipital lobe of the cerebrum. In addition, the cerebellum contains dopaminergic receptors for projection neurons from

the SNc, which means that cerebellar function is linked to the function of the basal ganglia (Wu & Hallett, 2013). Therefore, the proper functioning of the cerebellum relies on the proper functioning of the dopaminergic neurons of the SNc.

Basal Ganglia and Cerebellar Dysfunction in Parkinson's Disease

Parkinson's disease is clinically characterized by "resting tremor, slowness of movements, rigidity, gait disturbance and postural instability" (Wu & Hallett, 2013). This neurodegenerative disorder afflicts one-million Americans and is estimated to affect nearly seven to ten million people around the globe, with 60,000 new Parkinson's disease diagnoses every year in the U.S. alone (Beitz, 2014). Moreover, that number is likely to have increased; the prevalence of PD is estimated to increase, especially among the aging American population (Beitz, 2014).

Researchers study the pathophysiology of Parkinson's disease using radiotracers (Stoessl et al., 2014). One example of functional imaging that uses radiotracers is positron emission tomography (Stoessl et al., 2014). PET tracks membrane dopamine transporter, or DAT, allowing researchers to examine dopaminergic circuits and pathways in the brain (Stoessl et al., 2014). When PET with DAT is performed on patients with Parkinson's disease, the degradation of dopaminergic projections from the SNc is observed, and as discussed previously in this paper, this degradation leaves the individual unable to control movement (Akbarzadeh-t et al., 2021; Beitz, 2014; Chen et al., 2019; Stoessl et al., 2014). Moreover, the patient may also show cognitive and behavioral changes (Beitz, 2014). Typically, these symptoms surface when 50-70% of the substantia nigra's neurons have degenerated (Beitz, 2014). In addition, a decrease in dopamine firing by the SNc results in lowered RNA expression for the D1R and D3R receptors in the cerebellum, which implies that the dysfunction of the cerebellum and the dysfunction of the basal ganglia contributes to PD pathophysiology (Wu & Hallett, 2013).

Effect of Parkinson's Disease on Speech

It is common for patients with PD to have dysarthria—impaired speech—and dysphonia—impaired "production of vocal sounds" (Little et al., 2009). The muscles that control speech cannot be controlled properly due to basal ganglia dysfunction, resulting in more sluggish, unintelligible speech (Akbarzadeh-t et al., 2021). In order to analyze this speech, researchers extract acoustic features like jitter, shimmer, and fundamental frequency from speech samples of PD patients and regular patients (Akbarzadeh-t et al., 2021; Little et al., 2009). Researchers typically record the patients saying the vowel /a/ and extract data for computer algorithms (Akbarzadeh-t et al., 2021). This paper will focus on using such data to create an AI model that detects patterns in speech for a PD diagnosis.

Artificial Intelligence, Parkinson's Disease, and the Future

This project aimed to develop a cheap, cost-effective solution for detecting Parkinson's disease using speech data. The materials used had to be freely available technology in the developed world. This constraint was maintained; simply one computer, a WiFi connection, and the Internet were used to create an AI model for PD. Apart from these requisites, the model required virtually no cost to create. Anyone with an Internet connection could recreate or improve this model. Artificial intelligence allows anyone to tackle complex problems and develop solutions in a matter of days. The advantage of AI is its immense potential and capacity for implementation. There is a Pandora's box of possibilities, and knowing how to wield artificial intelligence to benefit the human species is a necessary skill in the 21st century, comparable to reading and writing for the centuries prior. Unfortunately, many are intimidated by the notion of artificial intelligence, even with most information on AI

being accessible on the Internet for free. Artificial intelligence is, at its core, relatively straightforward. An AI model learns patterns using an algorithm or a dataset and then predicts outcomes based on those patterns (Janiesch et al., 2021). For PD, the model would learn patterns in speech data and then use those patterns to determine whether an individual has Parkinson's disease.

The speech data set was downloaded from the UC Irvine Machine Learning Repository for this paper. This dataset was compiled by Max Little of the University of Oxford, who worked with the National Centre for Voice and Speech, Denver, Colorado. Little et al. (2009) focused on the application of using "dysphonia measurements" as a diagnostic tool in telemedicine, and the team reached an accuracy of 91.4% using a "kernel support vector machine." It has been over ten years since the research was published, and it is now easier to make models that can learn patterns in data and apply them to new contexts. This paper will use the dataset of Little et al. (2009) to make a model that can discern between PD and non-PD patients with greater accuracy due to recent advancements in artificial intelligence. The dataset contains features like jitter, shimmer, and fundamental frequency, all of which are quantitative measures of dysphonia that can pinpoint those afflicted with PD (Little et al., 2009).

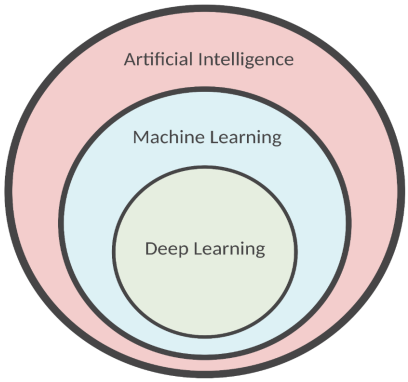


Figure 4. Deep learning is a form of machine learning, and machine learning is a form of artificial intelligence. Created using BioRender.com. Created and copyrighted by Ayush Tripathi.

The dilemma was whether to use machine learning (ML) or deep learning (DL) for the model. Machine learning is a subcategory of artificial intelligence, and deep learning is a subcategory of machine learning (Janiesch et al., 2021). The primary difference between "shallow" machine learning and deep learning is that in machine learning, it is necessary to tell the computer which features of the dataset to consider and the relationships between one feature and another (Heaton, 2017; Janiesch et al., 2021; Verdonck et al., 2021). In other words, "shallow" machine learning requires substantial feature engineering—manipulating features to make it easier for the model to discern patterns (Janiesch et al., 2021; Verdonck et al., 2021). In deep learning, this process is more automatic; the computer manipulates the features on its own, giving more weight and emphasis to certain features and less to others, though feature engineering can be done if needed (Heaton, 2017; Janiesch et al., 2021; Verdonck et al., 2021). Not requiring feature engineering is both an advantage and disadvantage; the programmer has one less task to attend to, but since the computer is performing feature engineering on its own, it is unknown how the features have been manipulated, akin to looking at a "black box" (Verdonck et al., 2021).

Given this comparison between ML and DL, a few propositions can be made:

1. Due to feature engineering, machine learning models are challenging to create but easier to explain and control (Verdonck et al., 2021).
2. Deep learning models are more straightforward for the programmer. However, it is more challenging to explain DL models because feature engineering is not done explicitly (Verdonck et al., 2021).

3. Advanced mathematics is the backbone of artificial intelligence, but for making basic models, knowing this math is unnecessary, and syntactic knowledge of the code will suffice.

For this paper, the most straightforward option was chosen: deep learning. The emphasis is not on pure theory or abstract minutiae but more on making a model as fast as possible, and deep learning was best for this purpose. Furthermore, Heaton (2017) suggests that support vector machines—the model type that Little et al. (2009) utilized–and DL neural networks "generally benefit from the same types of engineered features," which is further motivation to use deep learning. The methods for developing a successful DL model will be discussed next, so for those not well-versed in computer programming and artificial intelligence, it is possible to engage in self-study for several weeks and master the principles using the resources of the Internet. Doing so will make the following information about the model more relevant and comprehensible.

Developing the Model: Methods

The model was developed in a free online environment called Google Colaboratory. While creating the model in Colab, the model was sometimes too complex for the system to handle. At one point, the model was so substantial that even Colab's free TPU—an advanced processing unit for heavy computation—could not handle the model. As such, there was another constraint; the model would have to be executable within the limitations of the technology at hand. Undoubtedly more advanced hardware would benefit the model; however, the goal is to develop a superior quality model with as little cost as possible. It was thus necessary to carry out development with the materials at hand.

The first step in any deep learning endeavor is to obtain data. As mentioned, this data was obtained from the UCI Machine Learning Repository. The data was converted to a CSV file and uploaded to a Github repository[1] to get around another limitation. Usually, the data is converted to a CSV file and stored on the computer; then, the file path of the data is provided to the program. However, the file path is useless if the data cannot be stored on the computer or if another user accesses the code on the Internet without the file. Instead, to work around this, the program has code that takes a raw Github link and reads it directly. The raw Github link can be obtained by accessing the Github repository, opening the file with the data, and clicking "Raw." It is necessary to repaste the raw GitHub link periodically since the URL token may expire. Though this may seem more complicated than the file path method, the advantage of using a raw Github link is that anyone with access to the Github repository can run the model regardless of computer memory limitations.

The next step is to preprocess the data. First, the categorical data is dropped. Categorical data is all non-numeric, qualitative data. In this case, the only categorical data is the patient number, which is unnecessary. Once this data is dropped, only quantitative data is left. This data now needs to be divided. One critical step of machine learning and deep learning is to divide the data into two chunks: one for training the model and one for testing the model's accuracy (Nguyen et al., 2021). The model learns patterns from the training set and applies those patterns to the validation set. In addition, each training set and validation set comes in two forms: X and Y. X contains the features, and Y always contains the target value (Scikit-learn Developers, n.d.). For this model, the target value is the PD status of the patient. Training set X provides features for training, and training set Y provides target values for training; the model learns which features correlate to a particular outcome or target value. In the context of this paper, the model would associate specific values of jitter, shimmer, and fundamental frequency with the likelihood of a patient having Parkinson's disease. Before splitting the training and validation sets into X and Y, however, it is necessary to normalize the data. Normalizing the data means the same scale is applied to numbers across different features (Bhanja & Das, 2018). For example, if the

[1] Here is a link to the repository: https://github.com/ATripathiOG/ParkinsonDeepLearningExperimentation.git If this link does not work, navigate to https://github.com/ and search ParkinsonDeepLearningExperimentation.

values of one feature range from 20-40, and the values of another feature range from 30-60, both are put on a scale from 0-1. This technique makes it easier for the model to discern patterns in the data (Bhanja & Das, 2018). There are many methods for data normalization, and this PD model utilizes the min-max technique (Bhanja & Das, 2018).

After preprocessing the data, the model was created. It was programmed using the beginner friendly Tensorflow Keras. There are five components to the model:

1. Early_Stopping
2. Sequential steps
3. Optimizer and loss function
4. Fitting model to data
5. Validation loss and accuracy

These are intentionally included to improve the model and prevent overfitting and underfitting.

Overfitting vs. Underfitting

Overfitting and underfitting are intuitive yet essential concepts in machine learning and deep learning. Overfitting involves a model that is too specific to the training set (Allamy & Khan, 2015; Ying, 2019). Raw data always has random variation or noise; the data may follow a general trend, but not all data points will conform perfectly to the pattern (Gupta & Gupta, 2019; Ying, 2019). In overfitting, instead of finding the overall trend, the model becomes highly specific to the noise of the training set (Allamy & Khan, 2015; Ying, 2019). As a result, the model works exceptionally well with the training set, but because it is too specific, it will perform poorly with the validation set (Ying, 2019). The overfit model will minimize training accuracy loss but not validation accuracy loss (Prechelt, 2002; Ying, 2019). The ultimate goal is to minimize validation loss, and overfitting hinders this goal (Prechelt, 2002). Underfitting, another important concept, is the opposite of overfitting. In underfitting, the model has identified patterns that are too general to be effective (Ying, 2019). As a result, both validation and training loss are not minimized (Ying, 2019). The key to a competent model is moderation; the model should not be too specific or general to the training set, minimizing validation loss (Prechelt, 2002; Ying, 2019).

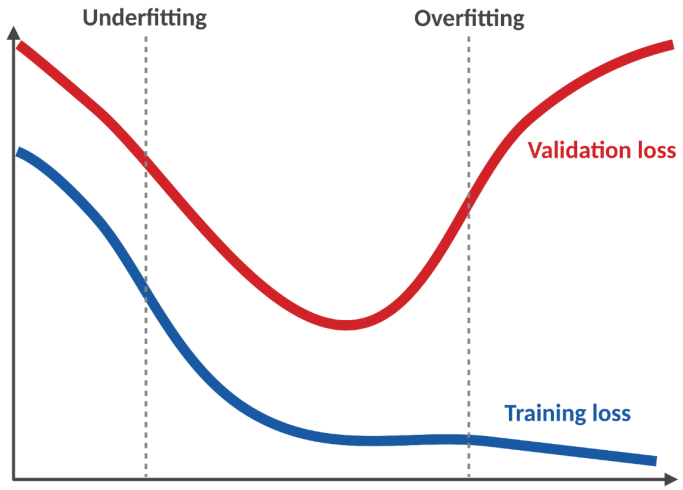


Figure 5. A graphical representation of underfitting and overfitting. Created using BioRender.com. Created and copyrighted by Ayush Tripathi.

Early_Stopping

"Early_Stopping" prevents overfitting (Allamy & Khan, 2015; Prechelt, 2002; Ying, 2019). If there is no substantial change in validation loss, "Early_Stopping" tells the model to stop running more trials (Allamy & Khan, 2015; Prechelt, 2002; Ying, 2019). There are several components of "Early_Stopping." If the "min_delta" argument in "Early_Stopping" is set to 0.00001, then a 0.00001 increase in validation accuracy is enough for training to continue (TensorFlow Developers, n.d.d). "Early_Stopping" will shut down the model if this 0.00001 increase is not met, but it has a tolerance. The "patience" argument gives "Early_Stopping" a tolerance (TensorFlow Developers, n.d.d). If "patience" is set to 20, the model can undergo 20 successive trials without improvement (TensorFlow Developers, n.d.d). Improvements in accuracy may not happen in successive trials; the threshold of 20 trials allows improvements to occur in nonconsecutive runs (TensorFlow Developers, n.d.d). Once an improvement occurs, the 20-trial threshold restarts (TensorFlow Developers, n.d.d). If 20 successive trials are completed without improvement, then "Early_Stopping" will shut down the model, and "restore_best_weights" will return the model to the trial with the best accuracy (Tensorflow Developers, n.d.d).

Sequential Steps of Model

Though it is a simplification, think of neural networks as a highly branched decision tree, with each decision given a certain weight (Grossi & Buscema, 2007). Neural networks have layers, and each layer has units. Each unit of a layer connects to units of the next layer (Grossi & Buscema, 2007). Thus, you have a highly branched decision tree called a neural network (Grossi & Buscema, 2007). This neural network can be created using a sequential TensorFlow model with multiple layers. In addition, the number of units in each layer can be expressed as a simple value within the parentheses of the corresponding layer (TensorFlow Developers, n.d.k). The “activation=‘relu’” argument transforms the values of each layer in a way that makes training more effective (Szandała, 2020; TensorFlow Developers, n.d.b). The “ReLU” function has a mathematical basis, but it is out of the scope of this text. The “input_shape” designates the number of inputs for the first layer of the model. For example, if there are 22 relevant columns in a dataset, there will be 22 inputs into the neural network (TensorFlow Developers, n.d.i; TensorFlow Developers, n.d.k). The “.dropout” layer prevents overfitting by randomly dropping some units in the layers to prevent the model from overanalyzing the training data (TensorFlow Developers, n.d.f). It is the equivalent of randomly applying weed killer to a vast expanse of grass. The weeds will be killed, but some helpful plants may also die from the chemicals.

Nevertheless, the benefits outweigh the costs. For example, a “.dropout” rate of 0.2 means that 20% of the units are dropped in the layer that occurs after the “.dropout” layer (TensorFlow Developers, n.d.f). Lastly, in the last step, “activation=sigmoid” takes the output and makes it a number between zero and one, with zero representing a 0% chance of having Parkinson’s disease and 1 representing a 100% chance of having Parkinson’s disease (Szandała, 2020; TensorFlow Developers, n.d.c). Again, note that the processes discussed in this paragraph can be mathematically demonstrated, but a more qualitative description is enough for simplicity.

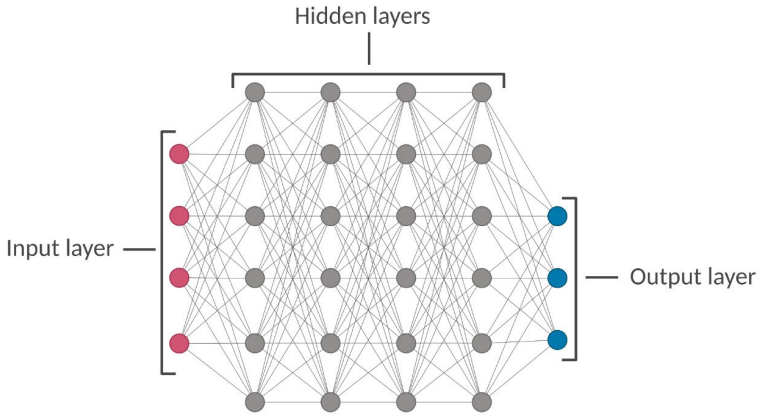


Figure 6. A conceptualization of neural networks. Created using BioRender.com. Created and copyrighted by Ayush Tripathi.

Optimizer and Loss Function

The optimizer function tweaks the neural network units (Kingma & Ba, 2014; TensorFlow Developers, n.d.j). The optimizer function adjusts the neural network to find the correct pattern to predict the target values using a set of given "features" (Kingma & Ba, 2014; TensorFlow Developers, n.d.j). This, too, has a mathematical basis that is out of scope for this text, but for simplicity, think of the optimizer like a whittler who carves wood. The wood comes uncarved, and the optimizer cuts the wood, shaping it in precisely the right way to satisfy the purpose of the wood. The loss function is easy to understand, too; it measures the discrepancy between the validation Y data (the actual target) and the predicted values of the target (TensorFlow Developers, n.d.g). Validation loss is necessary to see the accuracy of the model (TensorFlow Developers, n.d.g).

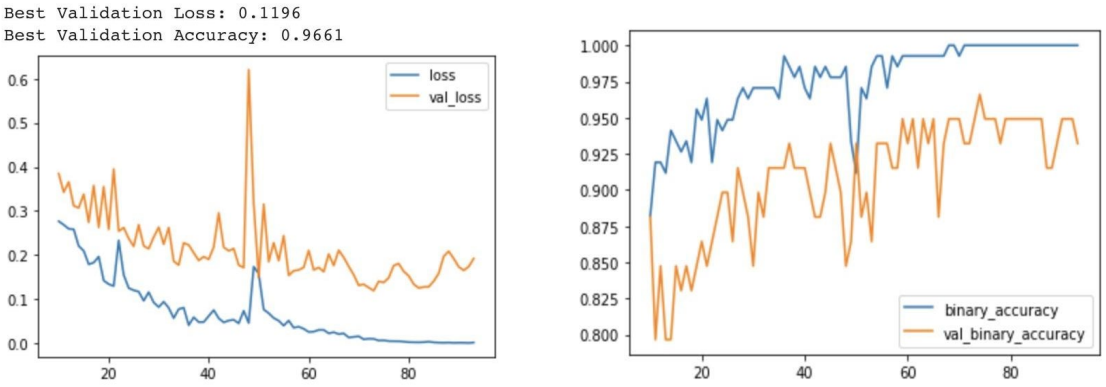
The "adam" optimizer and the "binary_crossentropy" loss function are suitable for the PD model developed in this paper (Kingma & Ba, 2014; TensorFlow Developers, n.d.g; TensorFlow Developers, n.d.j). As implied by Kingma & Ba (2014), there are a variety of optimizers and loss functions that are optimal for different scenarios. "Binary_accuracy," along with the loss function, is another metric that can be used to evaluate the correctness of this model; it is used when the target value (whether someone has Parkinson's or not) is characterized by a simple "yes" or "no" or a 0 or 1 (TensorFlow Developers, n.d.h).

Fitting Model to Data

In this step, the model is aligned with the dataset. The training and validation data are also identified—with the X and Y. The "batch_size" is listed as well; a "batch_size" of 50 means that the model will consider 50 data points at a time, process them, and then move on to the next 50 to process (TensorFlow Developers, n.d.e). Trial and error is necessary to determine the "batch_size." "Epochs" is how often the model will cycle through all the data (TensorFlow Developers, n.d.i). In this model, the "epochs" parameter was set to a large number since Allamy & Khan (2015), Prechelt (2002), and Ying (2019) indicated that "Early_Stopping" would stop the training if there was no substantial improvement in accuracy. The pandas library makes it possible to graph the validation loss and validation accuracy, and the graphs show the result of each epoch until the model was shut down by "Early_Stopping."

Testing the Model: Results

Multiple trials were conducted to test the validity of the model. The data was split randomly in each trial, and the model was run on that data. The following table contains the results of these trials. It is important to note that these were ten random trials, and doing another ten trials will entail similar--but not identical--results to the trials below.



Figures 7 and 8. A graph of validation loss and training loss, and a graph of training and validation accuracy. The graph shows the last trial in the table below. Created and copyrighted by Ayush Tripathi.

Tables 1 and 2. Validation accuracy varies from trial to trial, along with validation loss. Created and copyrighted by Ayush Tripathi.

Trial #	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6
Val. Accuracy	0.9492	0.8983	0.9492	0.9492	0.8983	0.9661
Val. Loss	0.1659	0.2340	0.1547	0.1580	0.1939	0.1243

Trial 7	Trial 8	Trial 9	Trial 10	Mean	St. Dev.	Mode
0.9492	0.9492	0.9322	0.9661	0.9407	0.02431	0.9492
0.1542	0.1336	0.1629	0.1196	0.1601	0.03389	N/A

Conclusion

When the model was run, on its best trial, it achieved nearly a 97% accuracy, meaning that if audio samples from 1000 patients were given to the neural network, it could theoretically diagnose around 970 patients correctly—while 30 would be misdiagnosed. On other runs, the model received slightly lower accuracies, but all accuracies fell approximately within the 90-97% range. The model, on average, had a 94% accuracy. It is also to be noted that throughout the process of making the model, very little domain knowledge of Parkinson's effect on speech was needed. The research done on the basal ganglia and speech was merely done to establish a relationship between speech and Parkinson's disease; it was entirely possible to make a successful model without knowing much about the disease itself, showing again how easy it is for non-experts to implement artificial

intelligence solutions to complex problems. In more complex or large-scale applications, a more rigorous understanding of artificial intelligence is needed, but that is not the aim of this paper; the goal is to show the ease of AI implementation. Artificial intelligence has continued to gain a foothold, and it is becoming easier to develop solutions that elevate the well-being of society. It is hoped that the reader will be encouraged to learn and ethically implement AI to solve issues in medicine and every aspect of life.

Acknowledgments

I want to thank my parents for their continuous support and encouragement. Also, I would like to thank Dr. Rajagopal Appavu, Coach Jothsna Kethar, Mrs. Smrutha Vikram, and associates for their mentorship. With the efforts of all these individuals, the project was able to come into being.

References

Akbarzadeh-t, M.-R., Azadi, H., Shoeibi, A., & Kobravi, H. (2021). Evaluating the effect of Parkinson's disease on jitter and shimmer speech features. *Advanced Biomedical Research*, *10*(1), 54. https://doi.org/10.4103/abr.abr_254_21

Allamy, H. K. J., & Khan, R. Z. (2015). Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). *Computer Science, Communication & Instrumentation Devices*. https://www.researchgate.net/publication/295198699_METHODS_TO_AVOID_OVER-FITTING_AND_UNDER-FITTING_IN_SUPERVISED_MACHINE_LEARNING_COMPARATIVE_STUDY

Beitz, J. M. (2014). Parkinson's disease: a review. *Frontiers in Bioscience*, *S6*(1), 65-74. https://doi.org/10.2741/S415

Bhanja, S., & Das, A. (2018). *Impact of data normalization on deep neural network for time series forecasting*. ArXiv. https://doi.org/10.48550/arXiv.1812.05519

Chen, Z., Li, G., & Liu, J. (2020). Autonomic dysfunction in Parkinson's disease: Implications for pathophysiology, diagnosis, and treatment. *Neurobiology of Disease*, *134*, 104700. https://doi.org/10.1016/j.nbd.2019.104700

[2]Chollet, F., & others. (2015). *Keras* [Computer software]. GitHub. https://github.com/fchollet/keras

[3]Cordasco, I. S., Lukasa, & Prewitt, N. (n.d.). *Requests* (Version 2.28.2) [Computer software]. https://pypi.org/project/requests/

Ellens, D. J., & Leventhal, D. K. (2013). Review: Electrophysiology of basal ganglia and cortex in models of Parkinson disease. *Journal of Parkinson's Disease*, *3*(3), 241-254. https://doi.org/10.3233/JPD-130204

Grossi, E., & Buscema, M. (2007). Introduction to artificial neural networks. *European Journal of Gastroenterology & Hepatology*, *19*(12), 1046-1054. https://doi.org/10.1097/MEG.0b013e3282f198a0

Gupta, S., & Gupta, A. (2019). Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, *161*, 466-474. https://doi.org/10.1016/j.procs.2019.11.146

Heaton, J. (2017). *An empirical analysis of feature engineering for predictive modeling*. ArXiv. https://doi.org/10.48550/arXiv.1701.07852

[2] Keras is a popular TensorFlow API that was used for the deep learning model.

[3] Requests is a popular Python module (does not come with default Python) used for HTTP commands. It was used to integrate the Parkinson's disease speech data from GitHub into the model.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3), 685-695. https://doi.org/10.1007/s12525-021-00475-2

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv*. https://doi.org/10.48550/arXiv.1412.6980

Kreitzer, A. C., & Malenka, R. C. (2008). Striatal plasticity and basal ganglia circuit function. *Neuron*, *60*(4), 543-554. https://doi.org/10.1016/j.neuron.2008.11.005

Lanciego, J. L., Luquin, N., & Obeso, J. A. (2012). Functional neuroanatomy of the basal ganglia. *Cold Spring Harbor Perspectives in Medicine*, *2*(12), a009621. https://doi.org/10.1101/cshperspect.a009621

Little, M.A., Mcsharry, P.E., Hunter, E.J., Spielman, J., & Ramig, L.O. (2009). Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. *IEEE Transactions on Biomedical Engineering*, *56*(4), 1015-1022. https://doi.org/10.1109/TBME.2008.2005954

[4]Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, … Xiaoqiang Zheng. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Retrieved January 10, 2023, from https://www.tensorflow.org/

Nguyen, Q. H., Ly, H.-B., Ho, L. S., Al-ansari, N., Le, H. Van, Tran, Van Q., Prakash, I., & Pham, B. T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, *2021*, 1-15. https://doi.org/10.1155/2021/4832864

[5]The Pandas Development Team. (n.d.). *Pandas* (Version 1.5.3) [Computer software]. https://zenodo.org/record/7658911#.Y_mkwS9Oni0

[6]Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2012, January 2). *Scikit-learn: Machine learning in python*. ArXiv. https://doi.org/10.48550/arXiv.1201.0490

Prechelt, L. (n.d.). Early stopping — but when? *Lecture Notes in Computer Science*, *1524*. https://link.springer.com/chapter/10.1007/3-540-49430-8_3

[7]Scikit-learn Developers. (n.d.). *Scikit-learn* (Version 1.2.1) [Computer software]. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Scikit-learn Developers. (n.d.). *3.1. Cross-validation: evaluating estimator performance*. Retrieved January 10, 2023, from https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation

Stoessl, A. J., Lehericy, S., & Strafella, A. P. (2014). Imaging insights into basal ganglia function, Parkinson's disease, and dystonia. *The Lancet*, *384*(9942), 532-544. https://doi.org/10.1016/S0140-6736(14)60041-6

Szandała, T. (2020). *Review and comparison of commonly used activation functions for deep neural networks*. ArXiv. https://doi.org/10.48550/arXiv.2010.09458

[8]TensorFlow Developers. (n.d.). *TensorFlow* (Version 2.11) [Computer software]. https://zenodo.org/record/7641790#.Y_ml-S9Oni0

TensorFlow Developers. (n.d.). *Tf.keras.activations.relu*. Retrieved January 22, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/activations/relu

[4] This is the TensorFlow white paper.

[5] Pandas is a library that was used in the code of the deep learning model.

[6] This is the paper that introduced Scikit-learn to the public. This was used for background knowledge.

[7] Scikit-learn is a library that was used in the code of the deep learning model.

[8] TensorFlow is a popular library for machine learning and deep learning. It was used in the code of the model.

TensorFlow Developers. (n.d.). *Tf.keras.activations.sigmoid*. Retrieved January 20, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/activations/sigmoid

TensorFlow Developers. (n.d.). *Tf.keras.callbacks.EarlyStopping*. Retrieved January 20, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

TensorFlow Developers. (n.d.). *Tf.keras.Input*. Retrieved January 22, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/Input

TensorFlow Developers. (n.d.). *Tf.keras.layers.Dropout*. Retrieved January 19, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout

TensorFlow Developers. (n.d.). *Tf.keras.losses.BinaryCrossentropy*. Retrieved January 22, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/losses/BinaryCrossentropy

TensorFlow Developers. (n.d.). *Tf.keras.metrics.BinaryAccuracy*. Retrieved January 20, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/metrics/BinaryAccuracy

TensorFlow Developers. (n.d.). *Tf.keras.Model*. Retrieved January 15, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/Model

TensorFlow Developers. (n.d.). *Tf.keras.optimizers.Adam*. Retrieved January 19, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam

TensorFlow Developers. (n.d.). *Tf.keras.Sequential*. Retrieved January 17, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/Sequential

[9]VanRossum, G., & Drake, F. L. (2010). *The Python language reference* (3rd ed.). Python Software Foundation.

Verdonck, T., Baesens, B., Óskarsdóttir, M., & Vanden Broucke, S. (2021). Special issue on feature engineering editorial. *Machine Learning*. https://doi.org/10.1007/s10994-021-06042-2

Wu, T., & Hallett, M. (2013). The cerebellum in Parkinson's disease. *Brain*, *136*(3), 696-709. https://doi.org/10.1093/brain/aws360

Yin, H. H. (2016). The basal ganglia in action. *The Neuroscientist*, *23*(3), 299-313. https://doi.org/10.1177/1073858416654115

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, *1168*, 022022. https://doi.org/10.1088/1742-6596/1168/2/022022

[9] Python, a popular programming language, and any integrated modules were used to create the model for this paper.