Lick the Toad: a web-based interface for collective sonification

Konstantinos Vasilakos [1]

[1]Center for Advanced Studies in Music – Istanbul Technical University
Istanbul, Turkey

konstantinos.vasilakos@gmail.com

Abstract. *Lick the Toad is an ongoing project developed as a web based interface that runs in modern browsers. It provides a custom made platform to collect user data accessed from mobile devices, such as smartphones, tablets etc. The system offers a tool for interactive collective sonification aiding the idea of networked music performance. It can be used in various contexts, such as onsite installation, interactive compositional tool, or for the distribution of raw data for live coding performances. The system embeds neural network capabilities for prediction purposes by using user input and outputs/targets alike. The inputs and the targets of the training processes can be adapted according to the needs of the use making it a versatile component for creative practice. It is developed as an open-source project and it works currently as a NodeJS application with plans for future deployment on remote server to support remote communication and interaction amongst distant users.*

1. Introduction

The idea of network as a means for real time distribution is not anew in the field of electronic music making. It goes back to the pioneering Telharmonium by an American inventor Thaddeus Cahill [Manning 2004] and also influenced by other attempts, such as Soemmerring's Musical Telegraph, and Tivadar Puskás' Telefonhírmondó, and the Musical Telegraph by Elisha Grey [Crab 2013]. What these all had in common, was that they all attempted to enact the transmission of sound generated by custom made electric instruments via hitherto available broadcasting technology e.g., telegraphy and landlines. One can argue that these can be seen as primitive attempts for music distribution deploying networking, which presumably paved the way to a new dimension of musical practice and artistic expression. It allowed for the decentralization of the sonic information by transmitting the performance to various ends using radio and wired networks for telecommunications. One can see an arguably far fetched analogy with much of the current practices in the broad landscape of Sonic Arts, such as live coding and laptop ensembles which are using ad hoc networks as a means to establish real time communication and distribution of various data amongst each other during their improvisation, such as the Hub's initial attempts for this paradigm of computer music networked performance [Gresham-Lancaster 1998, Collins and Escrivan Rincón 2011]. Of these, the Powerbooks Unplugged [1] and the Birmingham Ensemble for Electroacoustic Research (BEER) [2] [Wilson et al. 2014] are using custom made libraries for network communication, such as, the Republic (`https://github.com/musikinformatik/Republic`) and

[1]See `http://wertlos.org/pbup/`
[2]See `http://www.beast.bham.ac.uk/offspring/beer/`

Utopia (`https://github.com/muellmusik/Utopia`) respectively. Both running in SuperCollider (`https://supercollider.github.io/`) programming language as external libraries, and are used to provide coders an interaction platform in order to communicate with each other as a means to thrust their live coding performances, making ad hoc networks inherent part of their research and musical practice.

In the wake of this performance paradigm, it helped to address some questions about *telepresence* and *liveness* via remote and distributed performers [Rohrhuber et al. 2007] using network communication. Besides the distribution of various data types through ad hoc local network some other examples using audio signals via the internet enacting remote communication include BEER's "Recalibrated" performance of a live coding piece entitled *Pea Stew* [BEER 2021], an adaptation of Nick Collins' Pea Soup [Collins 1974], and the Symphony in Blue 2.0 by the Istanbul Coding Ensemble [3], a live coding performance based in Kamran Ince's composition entitled Symphony in Blue (2012) [4]. While the former piece involves the feedback of audio signals amongst remote coders the latter uses piano signal for machine listening utilization both locally and remotely using SonoBus (`https://sonobus.net`), a low latency state of the art audio transmission software. Lastly, besides network being used as a medium to distribute audio information and/or data, some works have used directly its inherent elements, for example the Network Ensemble has used real time sound representations of raw packets from WiFi networks [Smith and Tacchini 2017], and Chris Chafe's music work entitled Ping [Chafe 2001] used the network's ping latency information to modulate sound synthesis parameters.

1.1. Sonification and auditory display

Sonification is the practice of rendering data to sound [Hermann et al. 2011, Worrall 2019]. Real time streams of data or static data sets can be used as raw material to build wave forms or create associations between data streams and sound synthesis parameters. This simple yet fertile approach for experimentation has been widely explored in the field of the Sonic Arts and digital arts alike. Examples include [de Campo 2007] and the deployment of astronomic and environmental readings to particle collisions data [CPNAS 2017, Hill et al. 2017, TEDx Talks 2013, Sosby 2019, Mohon 2021].

In addition, sonification practice has been the workhorse for both artistic and educational outreach. Previous works by the author are demonstrated in [Vasilakos 2019] and [Vasilakos et al. 2020]. Given the technical affordances sonification can be explored as web applications and can run in browsers of mobile devices, such as smartphones, tablets and whatnot. This provides a personal exploration tool and at the same time it offers decentralized environments[5] which were previously unexplored due to the need of specialized background. Consequently, this opens up to non-technical backgrounds and the interested layman. Similarities in approach, includes IPSOS (`http://ipsos.web.cern.ch/`), a web-based application that runs on any modern browser and allows for the real-time sonification of data from the Large Hadron Collider, at CERN, in Switzerland.

[3]See `https://konvas.netlify.app/ice/`

[4]See `http://www.kamranince.com/html/compositions.php?id=189&page=`

[5]Decentralized environments refers to the standard model of laptop performances where the interaction relies on the single human computer relationship with the sonic generation.

1.2. Machine learning and the (Sonic) arts

Some examples that highlight using machine learning in music practice include Marije Baalman's "The Machine is Learning" [Baalman 2020], Sonami's and Fiebrink's [Fiebrink and Sonami 2020], and Sturm's work in the field [L. Sturm and Ben-Tal 2017]. Due to the development of machine learning modules in the browser, it allows for control without the need for specialized skills and interact with training and control of neural networks. This seems to be also observed by [Bernardo et al. 2020]. As Amershi et al. [Amershi et al. 2014] write,

> IML enables even users with little or no machine-learning expertise [to] steer machine-learning behaviors through low-cost trial and error or focused experimentation with inputs and outputs.

.

2. Lick the Toad

Lick the Toad offers an interface to interact with other users that are connected in an ad hoc network and share their positions to one another and at the same time train a model of a neural network using this activity. The interface runs in mobile devices and modern web browsers using a URL as seen in fig: 1. For an overview of the system see this link (`https://tinyurl.com/2jk5bvyf`). It is an open source project and can be found at this link (`https://github.com/KonVas/lick-the-toad`).

Besides of the training data generation the users can also engage in collective prediction process and collaborate with each other to build a real time work e.g., providing it is placed as an interactive sound installation. Due to the variability of the data it can be safely speculated that it may allow for a wide range of unpredictability of the musical outcome. This assumes the unlimited associations between data streams and sound, enabled by mappings that one has to establish with the sound generation modules in advance. The project is using SuperCollider but any platform such as Max/MSP (`https://cycling74.com`) and Pure Data (`https://puredata.info`) can be employed using Open Sound Control (OSC) communication. Thus the platform allows for active engagement with the sound generation instead of experiencing a musical work that was created with a digital musical instrument (e.g., as seen broadly in NIME community), which often may be viewed by the audience as a black box requiring programming dexterity and/or technical background in music technology.

2.1. Collection and training

The interface involves a single object which transmits to all connected users continuously its *X* and *Y* positions as float values relative to the screen borders of the device, see fig:1. While someone is interacting with the rest of the connected users they can see a string of the ID, which represents each user currently interacting. The system is able to collect the data and render it to a normalized range of 0.0 - 1.0 and start the training.

In addition to the *X* and *Y* values of the object, the user is able to add a target/output value. A snippet of the collected data prior of normalization can be seen in the *"data"* JSON snippet. The *ys* is the output value, that is the targeted value for prediction here represented as *frequency*. Numerous iterations and similar data sets using this template are

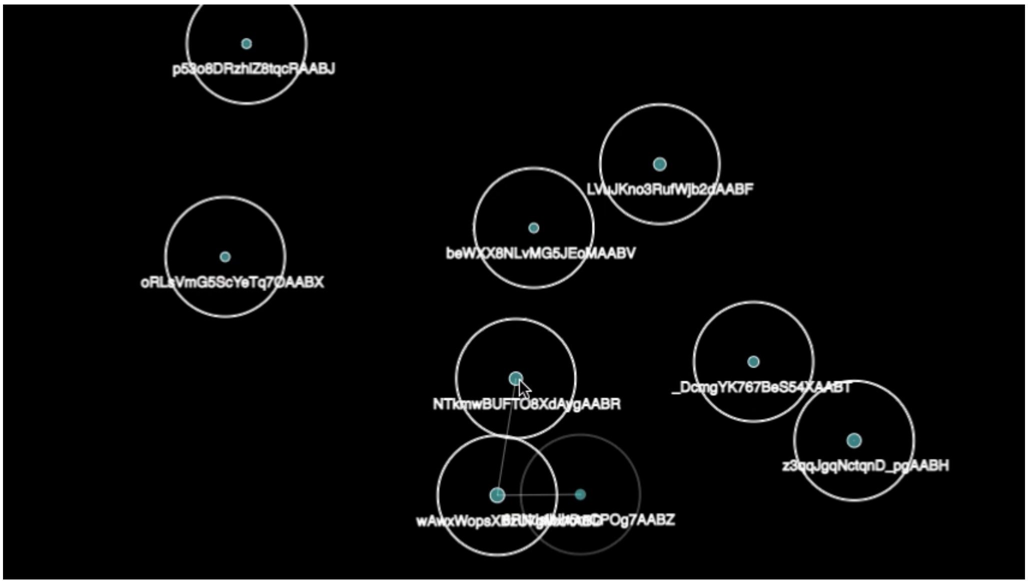

Figure 1. Interface for simultaneous user interaction.

stored in JSON format, the file can be saved locally in order to load the model and/or run the training process later. One can add other targets instead of *frequency* according to the needs providing that it follows the same format. The interface also creates visualizations of the receiving data using Berzier curves [6] as seen in fig:3.

```
{
"data":[
   {
      "xs":{
         "x":226.322645526962,
         "y":84.91814510458188
      },
      "ys":{
         "frequency":660
      }
   }
}
```

2.2. Training the model

The platform offers a minimal interface to start the various actions of the interface, such as collecting and training the model of the neural network. These actions are triggered using the native keyboard of the device. The neural network is implemented using the ML5 library (`https://learn.ml5js.org/#/reference/neural-network`), which offers a graph to monitor the training process as seen in fig:2.

It can be also monitored via the terminal window of the web page providing information about *loss, epochs* etc. of the training progress (see fig:4).

[6] `https://tinyurl.com/ykd6ywnu`

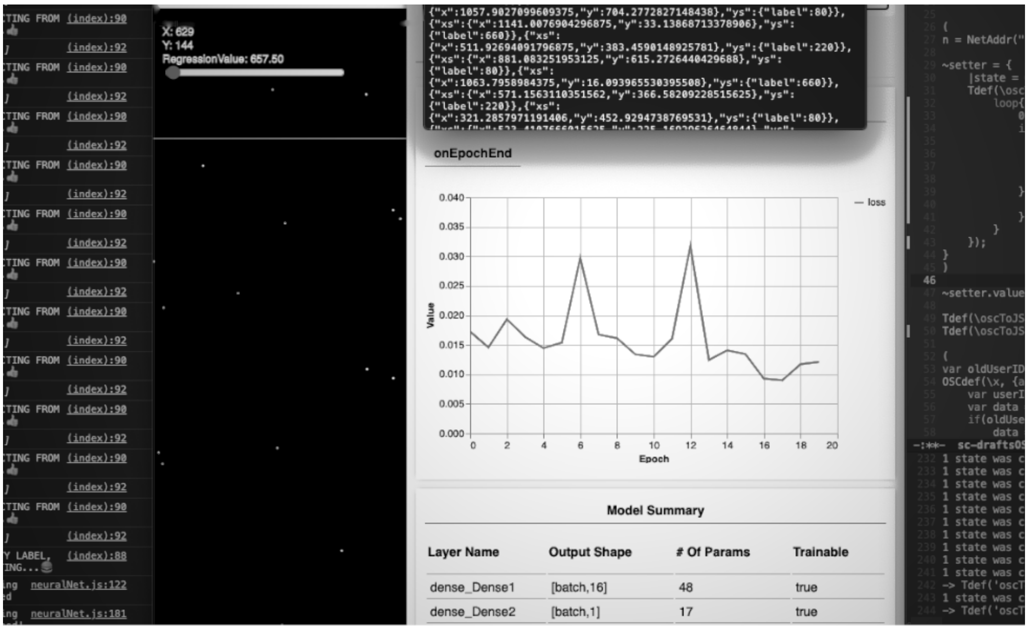

Figure 2. General overview of the interface.

2.3. Prediction and output

Once the system completes the training it starts generating prediction values based on the model, and triggering float values corresponding between the proximal points, for example, a user might enter some input data and declare it as *low*, *mid*, and *high*, their corresponding values (80, 220, 660) respectively; The output value will provide a continuous float number known as *regression value* (=x) between the targets, that is between *low* and *mid*, then this will be:

$$low \leq x \leq med$$

The selection is based on a cursor according to its relative *X, Y* positions. This is controlled manually or randomly using a Perlin noise operator [7] as shown in fig:5.

2.4. Mapping interaction

In addition to sending the regression value for sound mapping, the system also outputs the *X* and *Y* position of the cursor, that is, *X* and *Y* float numbers. For example, providing the cursor is between two points that are logged as *low* (=80.0) and *mid* (=220.0) the system will provide a value according to the proximity of these two and the same stream of the OSC message *X* and *Y* will be added. This format was found to be the most convenient for mapping the values to higher level musical parameters, such as frequency or duration to synthesise a sonic event.

The system can host diverse classification according to the nature of the input/output data, and thus generate a broad range of value range and mapping with sound sources accordingly. That allows the system to provide a versatile tool for experimentation in a sound synthesis context, instead of something that can be used only in a specific context and/or predetermined way which was devised by the author. For example, in live

[7] `https://p5js.org/reference/#/p5/noise`

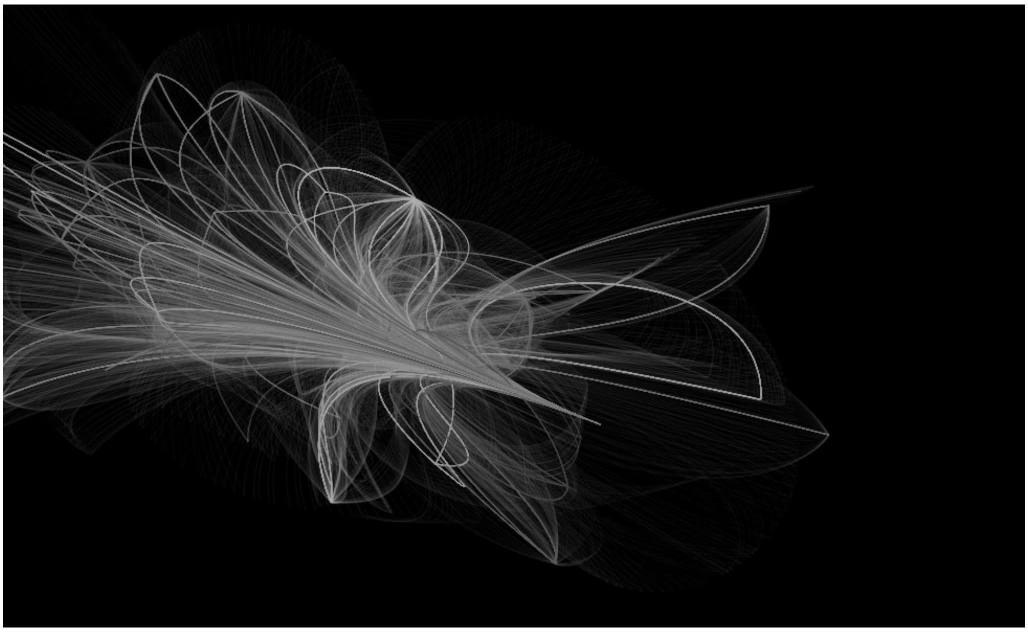

Figure 3. Visualization of user data using Berzier curves.

coding performances the system can acquire other data types to train the neural network. Successful tests using patterns to train the model generated from SuperCollider and used for live coding are also possible as seen in fig:6.

After the model is created it can be stored locally in the computer's hard disk and load again later for using it with other sound sources and software for musical creation.

3. Technical Details

TensorFlowJS (`https://www.tensorflow.org/js`), is a library for machine learning in the browser. ML5 is built on top of TensorFlow. It offers a higher level interface to implement machine learning in the browser including neural networks, pose recognition, and image prediction using pre-trained models offered by MobileNet (`https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet`). More information about the library is found at this link (`https://ml5js.org/`). To provide real time data distribution amongst users the project is using Socket.io (`https://socket.io`), a library that is designed to provide to applications real time communications, such as event handling. To communicate with SuperCollider the project uses OSC communication implemented with OSC.js (`https://github.com/colinbdclark/osc.js`) library. NodeJS, the runtime of JavaScript also provides a local web server, which is built on top of the interface. This is also tested on a Raspberry Pi 3 Model B (`https://www.raspberrypi.org/products/raspberry-pi-3-model-b/`). The visualizations are built in P5JS (`https://p5js.org/`), a link to the script can be found at this link (`https://github.com/KonVas/lick-the-toad`).

4. Creative Outcomes

Telematic art has been at the forefront of live coding practice and seems to provide a fertile solution to creating projects that focus on connecting remote user interactions, es-

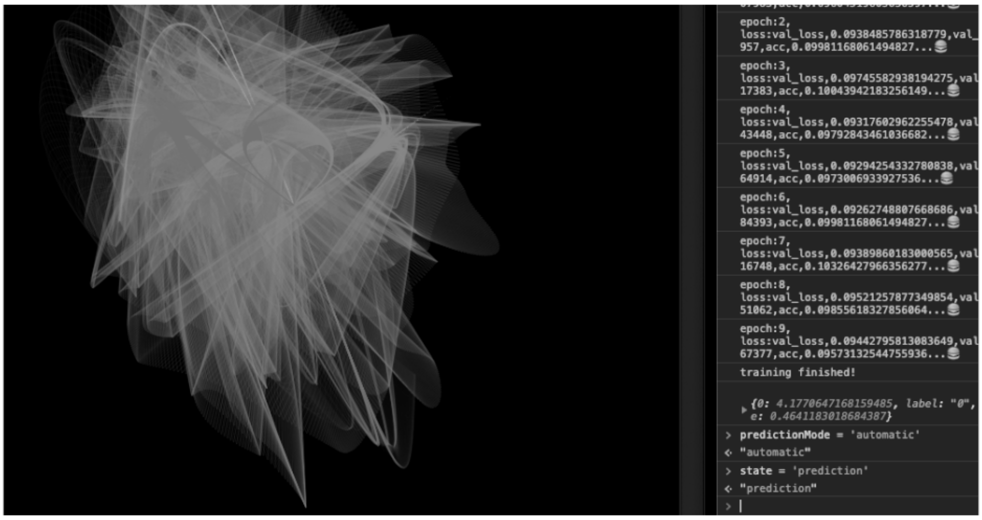

Figure 4. Training process and selection of prediction mode.

pecially in the wake of isolation and remoteness periods this may provide a useful tool for interaction and creative collaboration. While the project is in its initial stages, there are no performances that demonstrate the project's capabilities in such context. The system is a non-opinionated application that can be used in various ways that is, using data from logged users, or various streams generated by other applications and thus it can be presumed that it may be an interesting asset in the creative applications as follows.

4.1. Raw material for live coding

In the context of digital arts, live coding is the real time alteration of source code of running algorithms that generate sound or visuals [Collins et al. 2003, Nilson 2007, Zmölnig and Eckel 2007]. Live coding has been established as an emerging computer music paradigm with many enthusiasts using various tools and environments for improvising and tweaking source code in front of the audience. More information including tools and environments developed for live coding can be found at this link: (`https://toplap.org/wiki/Main_Page`). Previous works by the author in the field of live coding combining sonification practice includes [Vasilakos 2019, Vasilakos et al. 2020].

Besides the standard way of interacting with the native interface of the system it can be used to steer live coding performances. For example, providing decisions in the mapping between the incoming data and sound synthesis parameters. An exploration of this approach would include, generating the prediction values and streaming them to SuperCollider which it can adapt the data in additional functions that entail specific sounds to be triggered alike, e.g., *low, mid, high*, yet a very limited example for expressing the concept, as the targets can be adapted to much more detail and provide more sophisticated predictions based on the trained model. Then, the coder can improvise with the sound generation sources in SuperCollider in real time using all live coding techniques accordingly, (e.g., using JITLib in SC). During this performance one can explore live mapping strategies and its variants between the classification/prediction streams from the

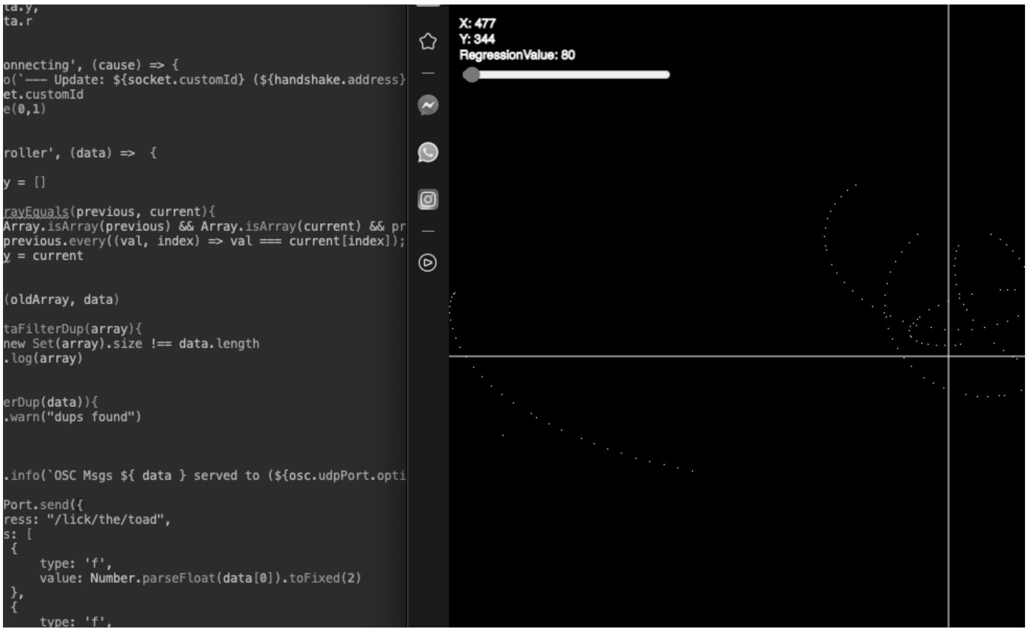

Figure 5. Selection of prediction using a cursor.

platform and the sound design algorithms generating an unpredictable musical outcome every time the system reiterates over the prediction data.

5. Future Directions

The application is at the stage of finalizing and running it via a local server in NodeJS. While it is an ongoing project, it is constantly adapted upon the requirements of projects. One of the future projects that the system may be used is with acoustic instrumentalists and train a model of the neural network using sensors for gestural information, and interpreted in various ways.

The primary goal of the project is to make the system accessible using a mobile device for online interactions. That means the users can interact from distant locations to train the system as well as interpret the data to create telematic sonification work. One possible way to implement this reciprocal interaction between training and sonifying amongst remote user(s) is to embed a remote server project developed by Serkan Sevilgen, a demo of this project can be seen at this link (`https://www.youtube.com/watch?v=uP1_Jp-sf0c`). For this it is required to be deployed and hosted on a remote server; this remains to be implemented in due course and finalization of other features will be implemented accordingly. Thus, next steps include the deployment of the platform as an online app which can host remote interactions amongst users across the internet and distribute data for various purposes, e.g., on site sound synthesis clients or live coders alike. Finally, further directions of this project will provide the combination of cosmetic and more eloquently appealing visualizations of the data, both in training and prediction modes. Finally, another future implementation which can be considered to expand on the creative outcomes of the system will provide the interaction between sound events and their visualization in real time, that is the sound events to be create real time graphics in another page of the interface.

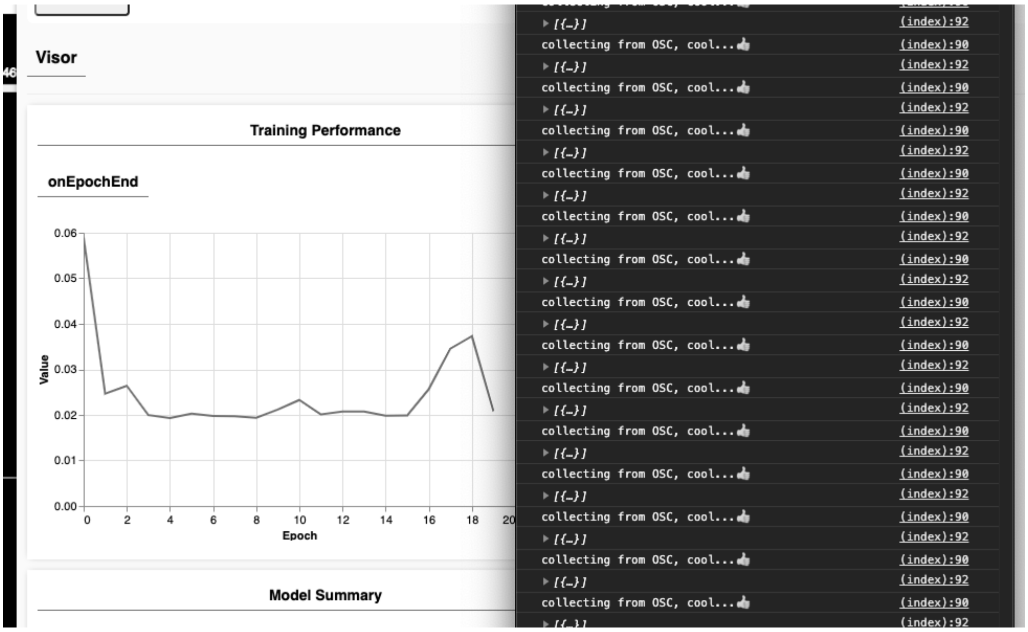

Figure 6. Collecting Open Sound Control (OSC) data generated in SuperCollider.

6. Conclusion

With the advent of machine learning technologies in the browser and its distributed collaborations it opens a brave new area for experimentation and musical research. Placing the audience in the forefront of the compositional epicenter instead of being a passive listener enables the collective interaction and real time communication amongst users and performers alike. This will provide for another level of exploration of creative nuances but also challenges for computer music practitioners and digital artists. Lick the Toad provides an interface to collect data from mobile devices and train a model to create further real time sonifications, which can be both triggered by the participants and/or used for live coding performances alike. So far, the system runs as a local environment in ad hoc network using NodeJS and SuperCollider for sonification purposes. The project has been proven to be quite flexible in terms of the collection of training data and thus it can be adapted to various projects, e.g., instruments and bystanders such as in the context of an on-site installation. Once the model is trained the interpretation can be mapped using OSC and SuperCollider as its sound engine. Further explorations, will allow for a fully decentralization of the data collection and run as a mobile application hosted and deployed on a remote server. This will allow for a wider interaction amongst users across the network.

References

Amershi, S., Cakmak, M., Knox, W. B., and Kulesza, T. (2014). Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, 35(4):105–120.

Baalman, M. (2020). the machine is learning.

BEER (2021). BEAST FEaST 2021: Concert 3.

Bernardo, F., Zbyszyński, M., Grierson, M., and Fiebrink, R. (2020). Designing and Evaluating the Usability of a Machine Learning API for Rapid Prototyping Music Technology. *Frontiers in Artificial Intelligence*, 3:13.

Chafe, C. (2001). Ping.

Collins, N. (1974). Pea Soup.

Collins, N. and Escrivan Rincón, J. d. (2011). *The Cambridge companion to electronic music*. Cambridge University Press, Cambridge. OCLC: 781183853.

Collins, N., McLEAN, A., Rohrhuber, J., and Ward, A. (2003). Live coding in laptop performance. *Organised Sound*, 8(3):321–330.

CPNAS (2017). Mark Ballora, Seeing with Your Ears: Data Sonification.

Crab, S. (2013). The 'Telharmonium' or 'Dynamophone' Thaddeus Cahill, USA 1897. https://120years.net/the-telharmonium-thaddeus-cahill-usa-1897/.

de Campo, A. (2007). A DATA SONIFICATION DESIGN SPACE MAP. In *Proceedings of the 2nd International Workshop on Interactive Sonification, York, UK*, page 4, York, UK. ISon.

Fiebrink, R. and Sonami, L. (2020). Reflections on Eight Years of Instrument Creation with Machine Learning. In Michon, R. and Schroeder, F., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 282–288, Birmingham, UK. Birmingham City University. ISSN: 2220-4806.

Gresham-Lancaster, S. (1998). The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music. *Leonardo Music Journal*, 8:39.

Hermann, T., Hunt, A., and Neuhoff, J. G., editors (2011). *The sonification handbook*. Logos Verlag, Berlin. OCLC: ocn771999159.

Hill, E., Cherston, J., Goldfarb, S., and Paradiso, J. (2017). ATLAS data sonification : a new interface for musical expression. In *Proceedings of 38th International Conference on High Energy Physics — PoS(ICHEP2016)*, page 1042, Chicago, USA. Sissa Medialab.

L. Sturm, B. and Ben-Tal, O. (2017). Taking the Models back to Music Practice: Evaluating Generative Transcription Models built using Deep Learning. *Journal of Creative Music Systems*, 2(1):1–29.

Manning, P. (2004). *Electronic and computer music*. Oxford University Press, Oxford ; New York, rev. and expanded ed edition.

Mohon, L. (2021). Data Turned Into Sounds of Stars, Galaxies, Black Holes.

Nilson, C. (2007). Live coding practice. In *Proceedings of the 7th international conference on New interfaces for musical expression - NIME '07*, page 112, New York, New York. ACM Press.

Rohrhuber, J., de Campo, A., Wieser, R., van Kampen, J.-K., Ho, E., and Hölzl, H. (2007). Purloined_letters. In *"Music in the Global Village Conference*, page 7, Műcsarnok Budapest. MV.

Smith, O. and Tacchini, F. (2017). *Network Ensemble Selected Network Studies*. Number SF0010 in The Strong of the Future. Rizosfera, Via Guido de Ruggiero, 6 – 42123, Reggio Emilia – Italia.

Sosby, M. (2019). Data and Music: What 50 Years of Exploring Our Moon Sounds Like.

TEDx Talks (2013). Listening to data from the Large Hadron Collider Lily Asquith |TEDxZurich.

Vasilakos, K. (2019). Emerging Dark Matter Through Diverse Sonification Practices. In *Proceedings of the International Music and Sciences Symposium*, pages 719–729, Istanbul. TMDK.

Vasilakos, K. n., Wilson, S., McCauley, T., Yeung, T. W., Margetson, E., and Khosravi Mardakheh, M. (2020). Sonification of High Energy Physics Data Using Live Coding and Web Based Interfaces. In Michon, R. and Schroeder, F., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 464–470, Birmingham, UK. Birmingham City University. ISSN: 2220-4806.

Wilson, S., Lorway, N., Coull, R., Vasilakos, K., and Moyers, T. (2014). Free as in BEER: Some Explorations into Structured Improvisation Using Networked Live-Coding Systems. *Computer Music Journal*, 38(1):54–64.

Worrall, D. (2019). *Sonification Design: From Data to Intelligible Soundfields*. Springer Publishing Company, Incorporated, 1st edition.

Zmölnig, I. and Eckel, G. (2007). Live coding: An overview. *International Computer Music Conference, ICMC 2007*, pages 295–298.