UNIVERSITY OF CALIFORNIA,
IRVINE

System, Component and Subcomponent Power Estimation

THESIS

Submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in ELECTRICAL AND COMPUTER ENGINEERING

by

Davit Hovhannisyan

Dissertation Committee:
Professor Fadi Kurdahi, Chair
Associate Professor Ahmed Eltawil
Assistant Professor Muhammad Al Faruque

2015

DEDICATION

To my parents for their unbelievable sacrifices.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENT

I would like to express the deepest appreciation to my committee chair, Professor Fadi Kurdahi, who has the attitude and the substance of a genius: he continually and convincingly conveyed a spirit of adventure in regard to research, and an excitement in regard to teaching. Without his guidance and persistent help, this dissertation would not have been possible.

I would like to thank my committee members, Professor Ahmed Eltawil and Professor Mohammad Al Faruque, for committing to be advisors and supporting me throughout these years.

I would like to thank all of my family and friends for love and support.

I thank the California Institute of Telecommunication and Information Technology (CALIT2) California Plug and Load Research Center for funding, mentorship, and guidance.

ABSTRACT OF THE DISSERTATION

System, Component and Subcomponent Power Estimation

By

Davit Hovhannisyan

Master of Science in Electrical and Computer Engineering

University of California, Irvine, 2015

Professor Fadi Kurdahi, Chair

This work focused on the power estimation of plug load devices, and in particular on Personal Computers. As a result, neural network classification estimated power with less than 5.4% errors. Study showed that internal performance counters properly described the overall system and the main component (CPU and GPU) power. Furthermore, neural networks model demonstrated higher precision on test data than the linear regression model.

INTRODUCTION

Personal Computers (PCs) are plug load devices, which have dynamic power consumption needs. State-of-the-art computing systems, such as PCs, use power reduction techniques. Some of these techniques focus on identifying wasteful operations and separation of such operations from mainstream. For example, when the PC is not in use and is ON, then it is wasting power, energy, and scrutinizing its lifespan. Hence, utilizing adaptive techniques increases efficiency. Some of these techniques utilize state identification for idle detection for deep sleep, or robust scaling of computational intensity and power consumption. Personal Computers (PCs) are complex systems composed of various components, and subcomponents. Typically found components in modern PCs are Power Supply, Motherboard, CPU, GPU, Hard Drive and DVD Drive. Each of these components' power consumption depends on the appropriation of the overall system. Layers of hardware and software such as power supplies, regulators, firmware, middleware, and operating systems operate Personal Computers. These layers determine the power and performance architecture, but the activity patterns of their users and running programs are the most significant source control. Thus, due to the complexity of the overall system (e.g. PC) and user's ever-changing usage patterns, it is not sufficient to estimate the overall power consumption with only rudimentary (e.g. peak frequency, voltage) metrics. Researches have shown that PCs are special purpose electrical appliances with unpredictable usage patterns. In 2005-2006, Thomas Beauvisage, from Orange Labs of France, carried out a study in computer usage in everyday life of 661 households with 1434 users at home for over 19 months. Among different findings, they identified that software preferences and usage intensity at the individual level are rather independent (Beauvisage, 2009). In another study conducted by Microsoft Research and Bell Laboratories found that Desktop PC is now a special purpose device, which they use only for specific activities such as working from home or online gaming. Thus, PC is an appliance such as a toaster or Microwave oven. Another study indicates that energy consumption of Desktop PCs varies over time from diversity of usage pattern impact on energy demands (Kawsar & Brush, 2013). Thus, inference on individual usage patterns from the general population of users may require resources beyond existing within computing system. Therefore, Desktop computers or PCs are appliances with stochastic usage pattern, and are an excellent model for study of real time power estimation for plug load devices.

Background

Power estimation and characterization are crucial for many endeavors, from competitive analyses to load characterization of the power grid. Moreover, it is important to characterize and estimate usage and demand for plug load devices from design to actual use. First attempts for accurate power estimation and characterization of digital systems are made in the design and testing stages by the very detailed knowledge of the inner workings of these systems. The usual solution is computationally expensive and time-consuming and is intended for use by system designers due to its detail of complexity. As final product moves from designers and manufacturers to consumers, the goals of estimation become less focused on detail and more concentrated on aggregate information. Moreover, as the challenge moves from the inner workings of a single unit to understand from a population of units, then more appropriate models of computation should be used. Thus, coarse black-box model based on power estimation methodologies, that rely on internal reporting of estimates with use of pervasive and observable features, can deliver estimates for very large deployments of computing devices, such as those plugged to the power grid.

Challenge with coarse black-box models identification of the optimal granularity or dimensionality of the model. The more detailed and refined these power states are and closer they correspond to actual hardware states more accurate power estimates become, at the cost of greater computational complexity. Therefore, it is of the utmost importance to identify those states that exert the most influence on power consumption without ending up with an inapplicable power model.

FIGURE 1: ACPI STATES

- G0 – Working
  - S0 – Processor Powered on (full on mode / connected standby mode)
    - C0 – Active mode
      - P0
      - P1
      - …
      - Pn
- G1 – Sleeping
  - S3 cold – Sleep – Suspend to Ram (STR)
  - S4 – Hibernate – Suspend To Disk (STD), Wakeup on PCH
- G2 – Soft Off
  - S5 – Soft Off – no power, Wakeup on PCH
- G3 – Mechanical Off

Some power management features are software interfaced in Personal Computers in the Advanced Configuration and Power Interface (ACPI), which help to save energy yet may be not possible to use for power estimation. ACPI standard defines multiple types of states, such as power, computational, and etc. For example, Intel's 4th generation CPUs' follow ACPI standards in order to save energy, and it is not designed to help with estimation. Intel Data sheet identified Global System (S), sleep (S), power (P) and package core (C) as separate types of states. The Figure 1 demonstrates ACPI states found in Intel 4th generation CPU's. It is important to point out, as stated in the documentation, that this structure may not be found in all processors or SKUs. Moreover, other than C0 state, other states consume very little or no power. Power States, P's, in C0 state do not guarantee that power and energy demand will be separated into coarse power ratings, thus, in the scope of estimation of power there will be not applied. However, it is known that such structure supports power and energy conservation (Hamady, F., Kayssi, A., Chehab, A., & Mansour, M.).

In work of (Park, Pasricha, Kurdahi, & Dutt, 2010), a layered approach is offered to deal with operational states and a granularity optimization technique is used to deal with the efficacy of the model. The work offers a layered approach to handle functionally accurate layer (Level 0), an instruction-level accurate layer (Level 1), a pipeline accurate layer (level 2), and a cycle-accurate or microarchitecture layer. According to their studies, the layered approach enables to deal with different planes of accuracy as needed. Instructions are the commands that are received by a processing unit. Therefore, it may be intuitive to assign power estimates to every instruction and calculate power by aggregation. This technique becomes far more complex when applied to the situation when such planes of complexity are increased.

An example of architectural complexity, the pipeline, in computing, is a form of operational complexity that may be implemented in hardware or software. Pipelines in their essence are very similar to ford assembly lines. In Figure 4, workers are working on the chassis of a vehicle each adding an effort to in a line. Due to specialization of each worker work finishes much quicker than if each worker had to complete the entire process by himself. Although pipelines simplify the work, they do not necessarily save power, or make it deterministic, moreover, they add another plane of complexity. Some of the added complexities are in part due to the added number of units and their operational states, as some may not be always deterministically operating.

Units are computational structures that directly deal with computing. These units are governed by instructions and pipeline. Thus, the work by Park and others offers an approach to simplify the interaction between these different abstractions to allow feasible mechanisms of power and energy estimation. This work tried to take advantage of this layered approach by dealing with all abstraction layers in a single layer of features.

Model

Hierarchical system model can be aggregate into lumped energy estimate or can report individual distributed energy estimates. The system model consists of Components, which may be devices, and are composed of Subcomponents. Subcomponents may have their own subcomponents as well. For example, in Home System consists of a PC (Personal Computer) with a CPU (Central Processing Unit), which has some states such as S1, S2, S3... These states represent power consumption of the Device. Thus, the formula (Equation 1) for power consumption at any given moment would depend on per device reported power consumption. Home system is an example of a system where this strategy may apply. Aggregating all component energy models allows having a system view on energy estimation. (Figure 2)

Equation 1: $$Total\_Power = \sum P_i$$

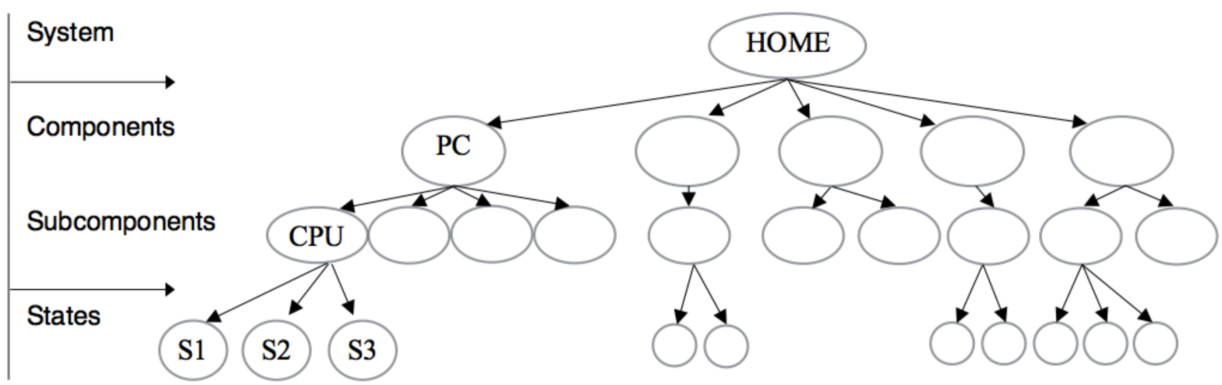
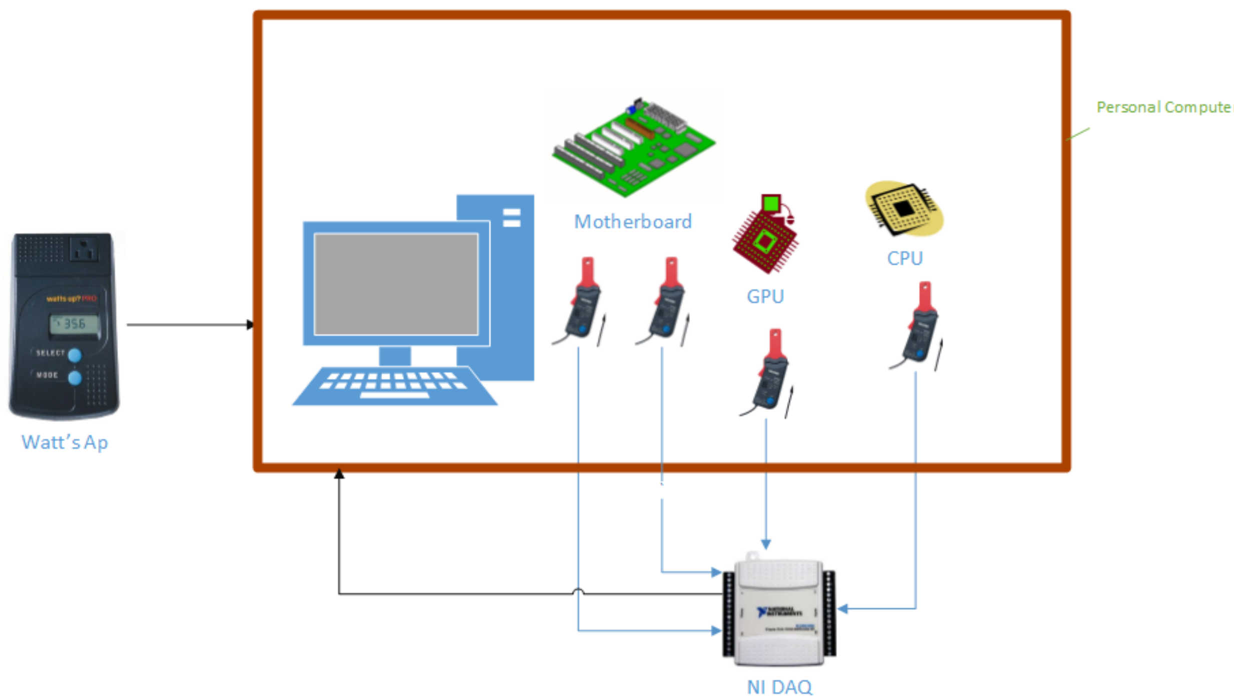


FIGURE 2: HIERARCHICAL MODEL

Procedures

Methodology uses three complementary procedures. The first procedure uses external metering for overall system energy consumption and calibration. Second procedure utilizes internal metering of internal components for energy consumption estimation and calibration. Third procedure uses a statistical learning model based on performance metrics or "performance counters" that are prevalent in different parts of the overall system, to estimate energy

FIGURE 3: EXPERIMENTAL SETUP DIAGRAM



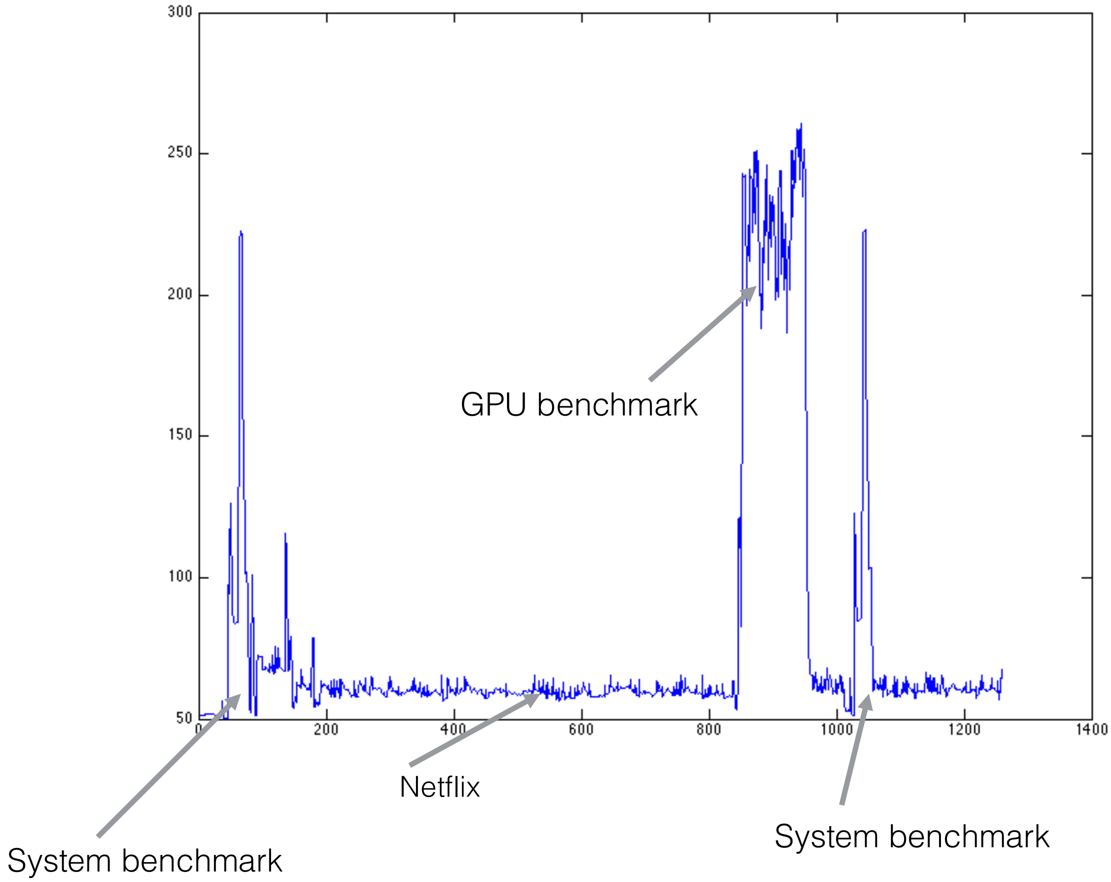
consumption for the system and its components. In this work, we discuss complementary set of machine learning techniques for accurate electrical energy estimation.

In Figure 3, demonstrates the setup of the physical system. Diagram demonstrates connection of different metering units and computer. These units measure overall PC and Components power measurements. The system is powered thru "Watt's Ap" meter, which serves as a power meter for the overall system. The "Watt's Ap" meter has a limitation of internal logging for approximately 32,000 samples. The study used the minimum sampling period of 1 second.

Current clamps were used to measure current flowing thru wires connecting power supply of the PC to Motherboards appropriate sockets. Measured current is used to measure power consumption of a particular socket, because the voltage is regulated by the power supply. Procedures in the study included identifying applications and programs, which were used as benchmarks, such as those mentioned in Appendix 1. These programs enable exposure of power states of the system and its individual components. Demonstration of the experiment that exposed the underlying system is in Figure 4.

Next step ensures that all logging tools and devices log the data with respect to the same time reference. Because multiple measuring units were used, a convex optimization technique is applied to properly align their measurements with software internal performance metrics. Essentially asynchronous metering related problem was solved with use optimization technique for alignment.

FIGURE 4: POWER VARIATION OVER DIFFERENT APPLICATIONS



In order to correct identified misalignment, alignment study was done by using mean square error cost function with respect to different arrangements in data streams of inputs and outputs. The optimization heuristic was to align sequences of data streams in such way that would enable lowest cost.

In the Figure 5, convex graph demonstrates convex minima at which a correction needs performed on input and output data streams. The low point in the graph shows that at that point cost function, which is related to the error of estimation is the lowest.

This technique allows use of inexpensive asynchronous metering systems, which do not share a common reference with one another. Figure 6 and 7 demonstrate different fits of estimation under different time sequence arrangements.

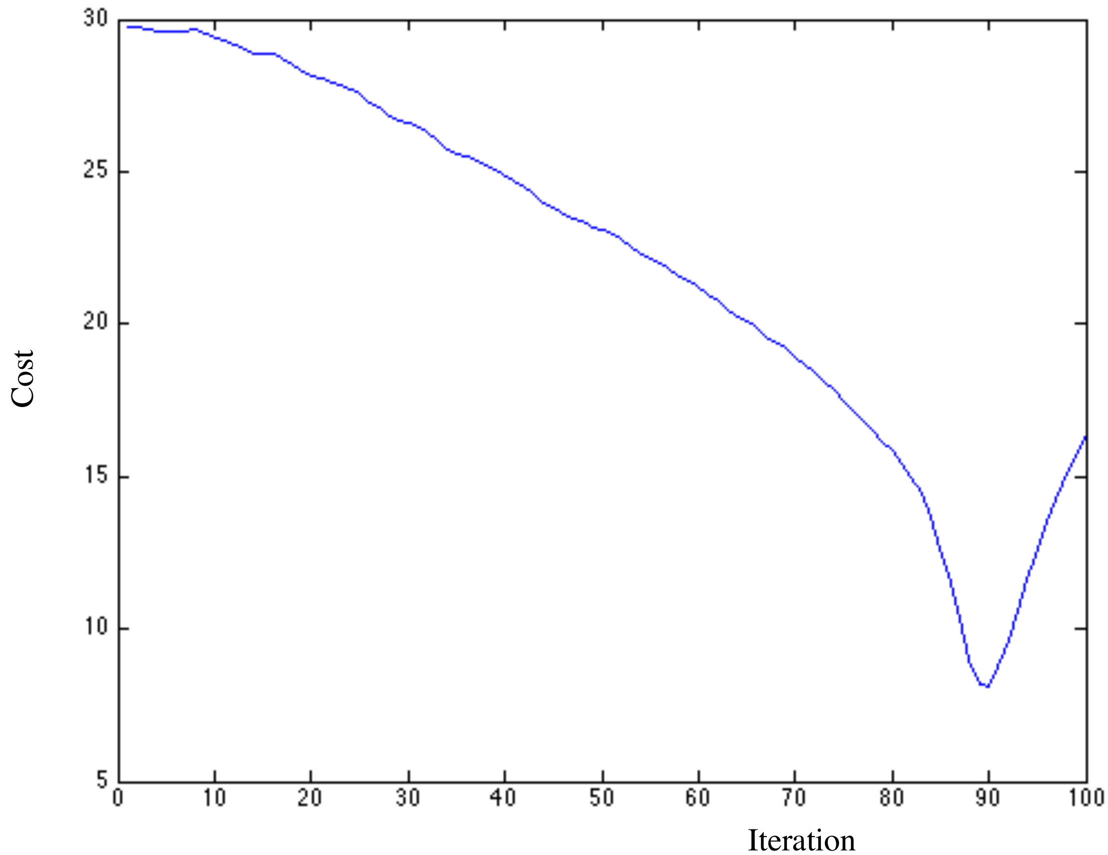


FIGURE 5: CONVEX MINIMA ALIGNMENT OF TWO INDEPENDENT METERS (90 POINT DRIFT NOTICED)

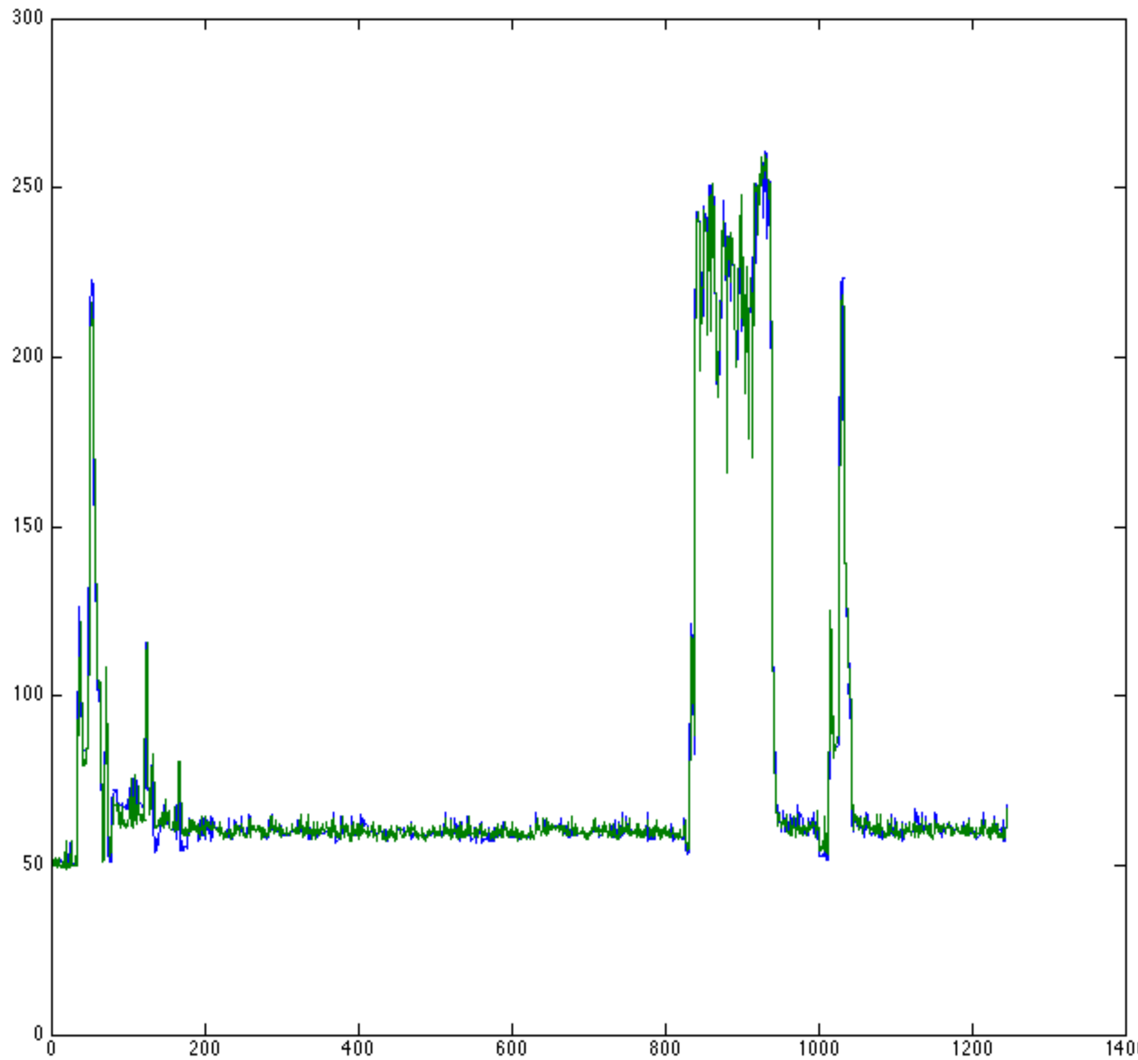


FIGURE 6: REGRESSION FIT FOR PERSONAL COMPUTER (PC)

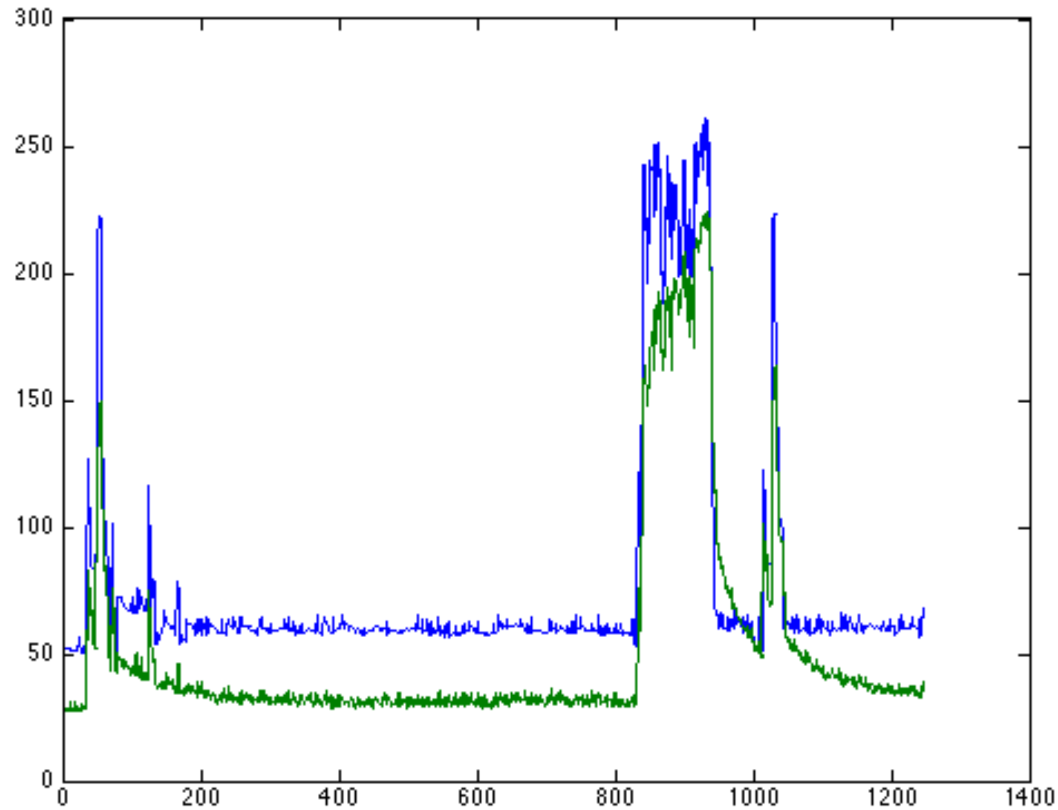


FIGURE 7: NOT ALIGNED DATA REGRESSION RESULTS

After proper alignment is done on data streams, supervised learning algorithms can be applied to estimate energy consumption with linear regression estimation and neural network classification.

Although, performance counters are intended to provide information about performance of different software and software layers, research used performance counters as features for statistical inference of power estimation by machine learning techniques. Software may also use the counters to store information on system resources allocated to adjust its strategies of operation. (Microsoft Corporation).

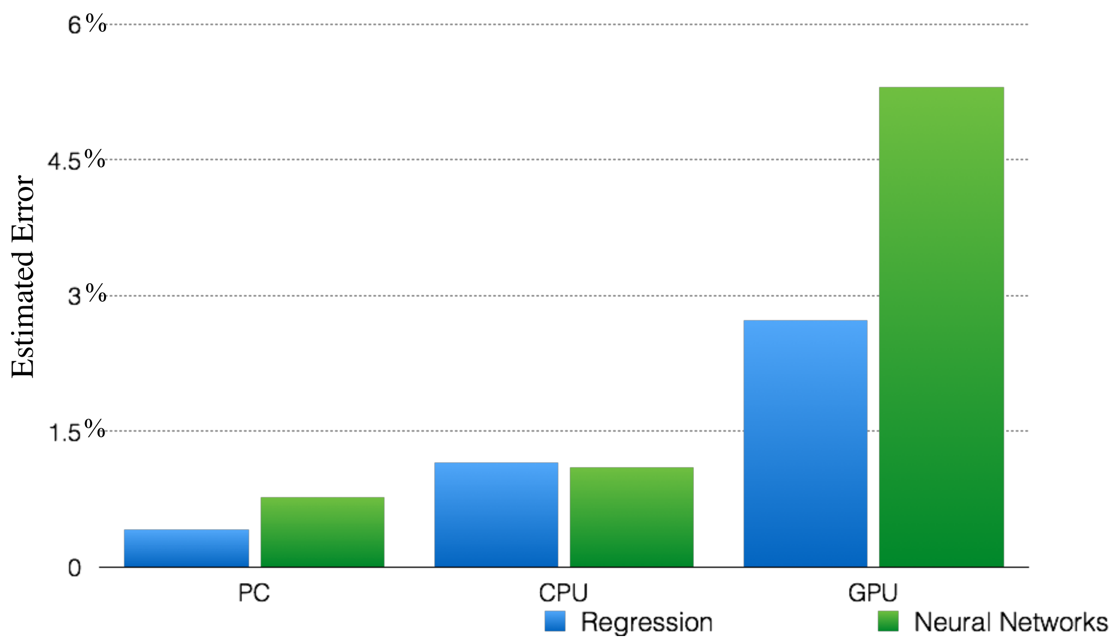
In order to exploit the data from performance counters, two different strategies of estimation were used; respectively, regression and neural networks models of supervised machine learning used linear and nonlinear transformations to estimate consuming power and energy.

Machine Learning is a scientific field, which uses data to construct learning algorithms. Supervised learning is a process by which known examples of correspondence is used to make inferences.

Results

Results shown in Figure 8 demonstrate that the mean estimation error in percent is usually higher with Neural Networks in comparison to Linear Regression, however, when a test sample is used, as shown in Figure 9 estimates are more accurate with Neural Network model. It is important to mention that linear regression is computationally less intensive. Both algorithms can be implemented on-line, meaning that the data does not have to be present from the very beginning and it can be added as more data is available. Accuracy of supervised learning algorithms may improve with more data. This features allows for continues self-calibration.

Figure 8: Accuracy comparison of Regression and Neural Network for PC, CPU, and GPU



Mini-Bucket batch processing technique and others allow neural networks model to be refined by processing new data after an initial model is formed. Thus, a particular model of a device, such as PC, may have specific generic model assigned to it that will offer generally accurate results, however, due to variability in between different devices of the same model, a self calibration can be done to compensate for intrinsic differences.

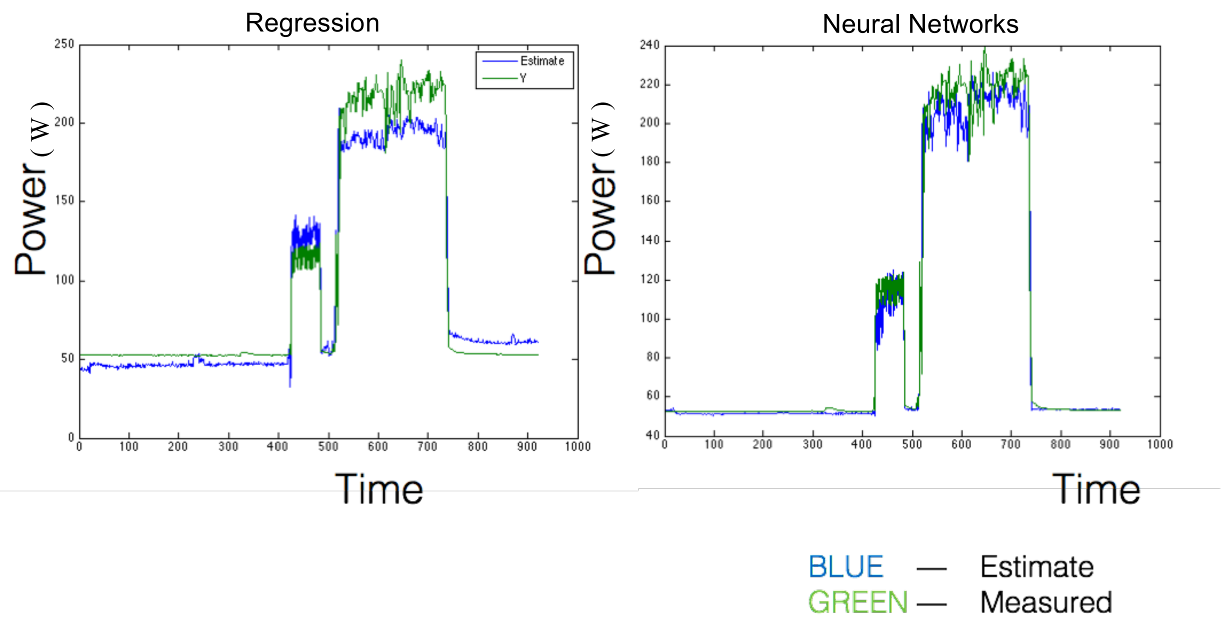


FIGURE 9: REGRESSION AND NEURAL NETWORKS PERFORMANCE COMPARISON

In Figure 10, results show that there is high power consumption variation between different operational states. For example, the range of power varies of PC between 50-290 Watts while range of the CPU varies up to 50 Watts. Results demonstrate high power variation over different patterns.

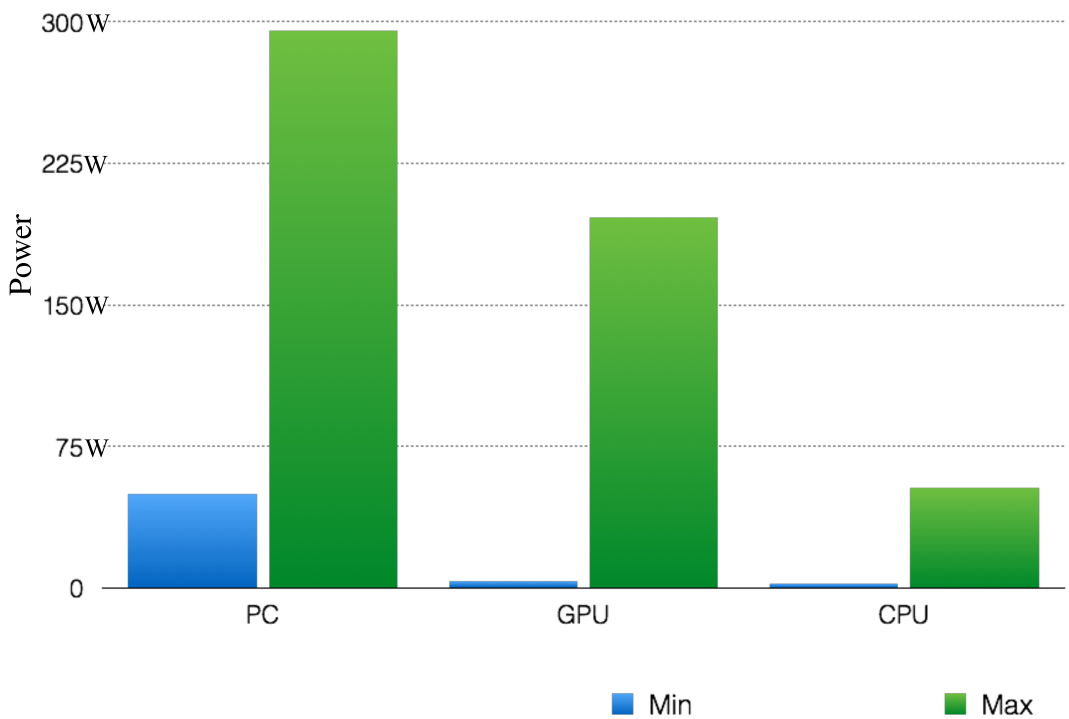
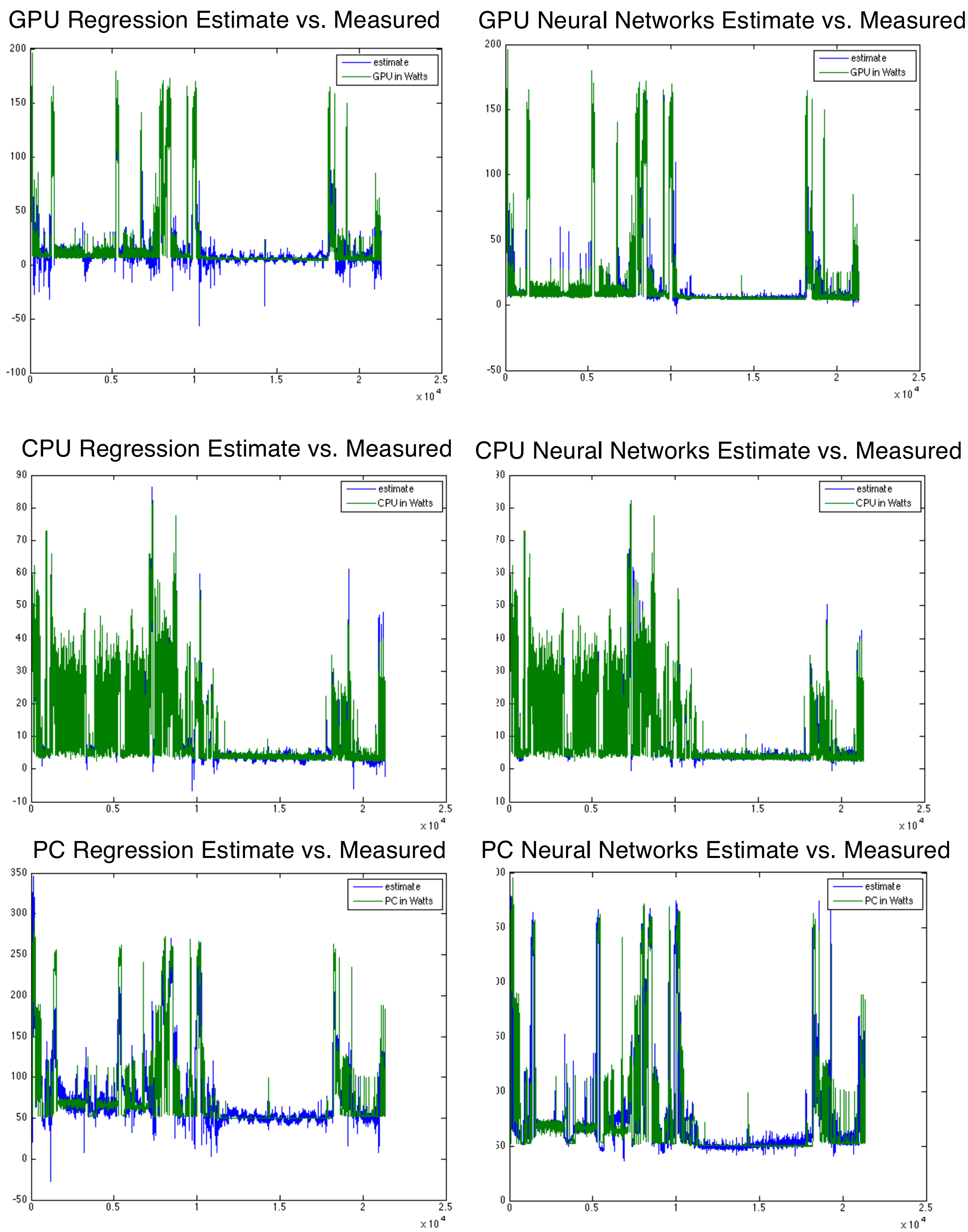


FIGURE 10: POWER VARIATIONS OBSERVED

FIGURE 11: RESULTS FOR REGRESSION AND NEURAL NETWORKS (VERTICAL AXIS AVERAGE POWER IN WATTS, HORIZONTAL AXIS SAMPLES FOR EVERY TWO SECONDS)



Results demonstrate accurate results with errors 0.42-5.3% in terms electrical energy. In Figure, study results are demonstrated for PC, GPU and CPU. These results are spread between regression estimation and neural networks classification (Figure 11). Each individual graph shows fitness between measured and estimated power values. The results can be further analyzed by electrical energy used. Electrical energy in each study is the sum of every point in the graph or the area under the curve. Thus, the total energy estimated over the study can be compared with measured by comparing areas under the curves. However, to have more appropriate results, comparison should be done on a test data, which was not included in the learning model. Table 1 demonstrates results on test data where error varies between 0.42%-5.3%.

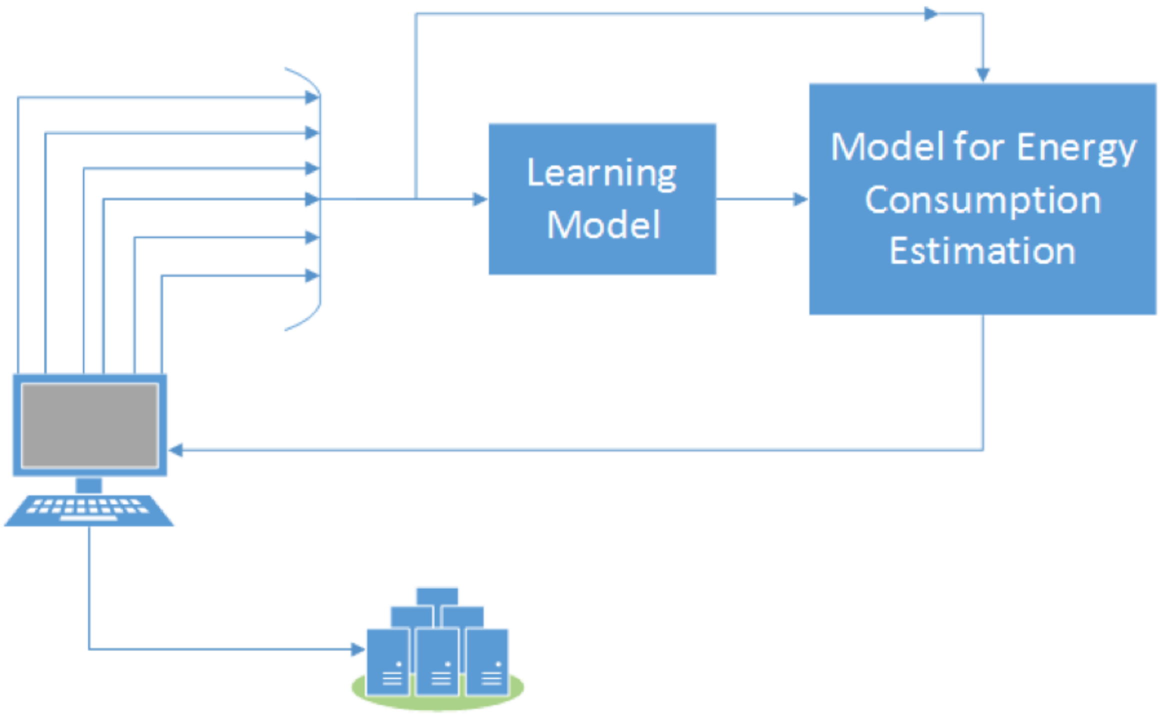
TABLE 1: ESTIMATION ERROR

	PC	CPU	GPU
Regression	0.42 %	1.16 %	2.73 %
Neural Networks	0.77 %	1.1 %	5.3 %

Internal energy reporting offers insights for power consumption, allows visualization, enabling more judicious power usage behavior for consumers to discover subtle sources of wasting components and devices.

In ERROR! REFERENCE SOURCE NOT FOUND.summary of steps taken to report energy estimation is illustrated. Performance metrics taken from the PC are then processed by one of two, regression, and neural networks, learning models, which produce estimates and are sent to aggregating and visualizing system. Then the data can be visualized from an online database by using advanced visualization techniques, which can conform to users needs.

FIGURE 12: PERFORMANCE COUNTERS USED TO DERIVE A LEARNING MODEL AND USE IT FOR ENERGY CONSUMPTION ESTIMATION AND REPORTING.



Conclusion and Future Work

Once estimated, data can be reported to visualization tools that will allow non-experts visualize power consumption patterns. The insights attained from power consumption measurement and visualization may enable more judicious power usage behavior and allow customers to discover subtle sources of power waste, such as long-running unused subcomponents of computing devices. The end goal is to make power consumption to closely track useful activity expected by users and consumers.

The results of this work suggest that an easy-to-use software can be developed that can be used to study personal computers with their respective configuration post production time which will enable users to know beforehand how much energy a particular configuration uses for particular software or benchmarks. The suggested software has already been designed and can soon be available.

Graphical user interface in Figure will allow users review their energy consumption or redo the experimental study with only few additional hardware component used in the study itself. This will allow greater transparency for energy consumption of systems and will allow users take control of their energy budget. The cost of such hardware setup may vary between \$100-\$700, depending on how many metering units are used.

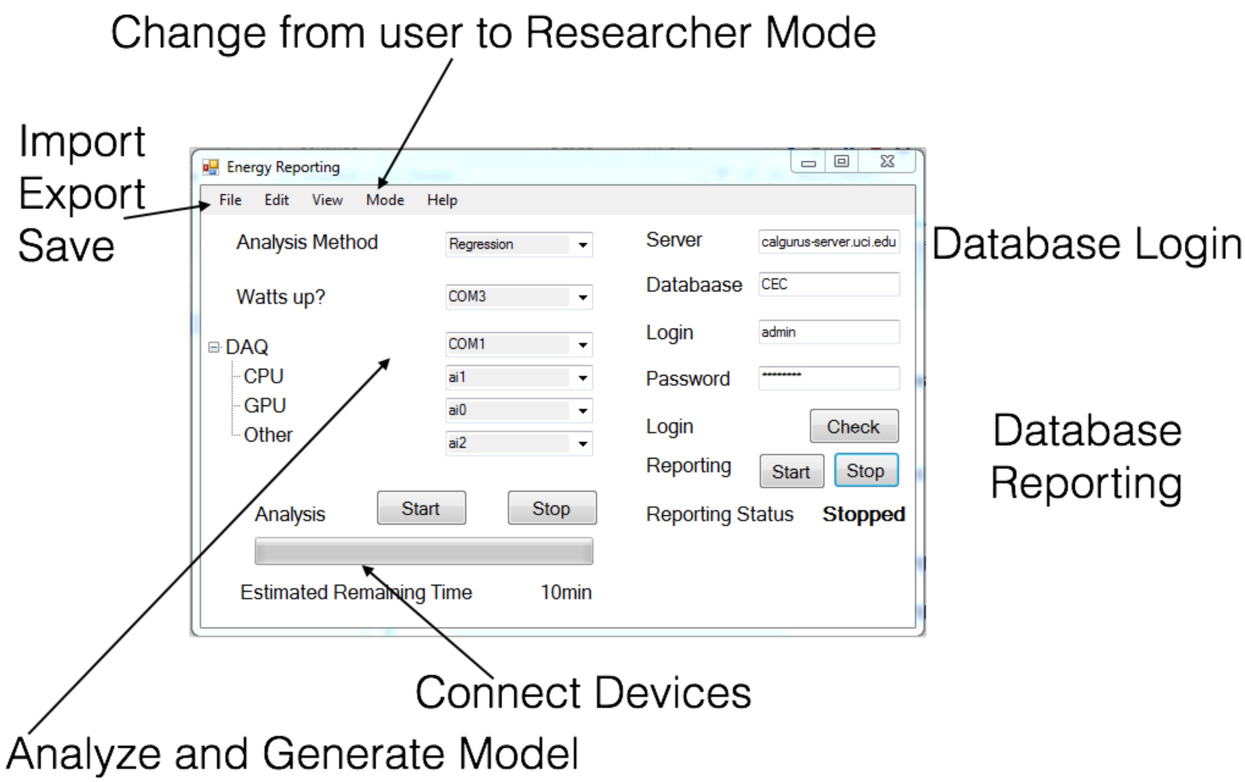


FIGURE 13: SOFTWARE FOR ENERGY ESTIMATION MODELING AND REPORTING

Results of this study show that it is possible to give expert designer understanding of components and subcomponents of home devices as sophisticated as personal computers. Energy estimation and reporting can increase transparency and awareness of users with little or no background in system power consumption and their needs as opposed to leaving them with guess work and trust towards sales associates who themselves may not have appropriate understandings of possible energy waste generated by some components and software.

Bibliography

Beauvisage, T. (2009). Computer Usage in Daily Life. Boston, MA, USA: CHI 2009.

Hamady, F., Kayssi, A., Chehab, A., & Mansour, M. (2012, May). Evaluation of Low-Power Computing when Operating on Subsets of Multicore Processors. J Sign Process Syst .

Intel Corporation. Mobile 4th Generation Intel® CoreTM Processor Family, Mobile Intel® Pentium® Processor Family, and Mobile Intel® Celeron® Processor Family .

Kawsar, F., & Brush, B. A. (2013). Home Computing Unplugged: Why, Where and When People Use Different Connected Devices at Home. Zurich, Switzerland: UbiComp'13.

Park, Y.-H., Pasricha, S., Kurdahi, F. J., & Dutt, N. (2010). A Multi-Granularity Power Modeling Methodology for Embedded Processors. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , 19 (4), 668 - 681.

Microsoft Corporation. About Performance Counters. (n.d.). Retrieved January 6, 2015, from http://msdn.microsoft.com/en-us/library/windows/desktop/aa371643(v=vs.85).aspx

Microsoft Corporation. Performance Counters in the .NET Framework. (n.d.). Retrieved January 6, 2015, from http://msdn.microsoft.com/en-us/library/w8f5kw2e(v=vs.110).aspx

Clemson University. Retrieved January 6, 2015, from http://www.clemson.edu/caah/history/FacultyPages/PamMack/lec122/ford.htm

Appendix 1: Tools and Benchmarks

Performance Benchmarks

- Intel Extreme Tuning Utility
- PC-Mark 7
- Unigine Valley Benchmark
- Unigine Heaven Benchmark
- SiSoftware Sandra

Programs as Benchmarks

- OpenOffice
- Netflix streaming
- Amazon Instant streaming
- Microsoft Visual Studio
- Eclipse IDE

Logging tools:

- For logging data from PC, three software tools were used: Intel Power Gadget, Asus GPU Tweaker, and Windows Performance Monitor.
- For individual components power consumption logging current clumps were used with National Instruments DAQ unit and NI DataLogger utility from the NIMxBase software suit.
- For system power consumption “Watt’s Ap” power meter was used.

- For deduction of power in different components study used current clamps. The current clamps used were made by PicoTech, and the model was PP218, which is a 60 A AC/DC Current Clamp.

Appendix 2: Methods

Regression Method

After alignment correction, data is ready to be used to produce regression and neural networks models. In regression data resulted fit is illustrated which is an improvement over **Error! Reference source not found.**. Similarly, Figure 14, Figure 15, Figure 16 and Figure 17 respectively, demonstrate the fits for CPU – Central Processing Unit, GPU – Graphical Processing Unit, Motherboard and PC overall power consumption. Note, in green is for the measured data and blue is for the estimated.

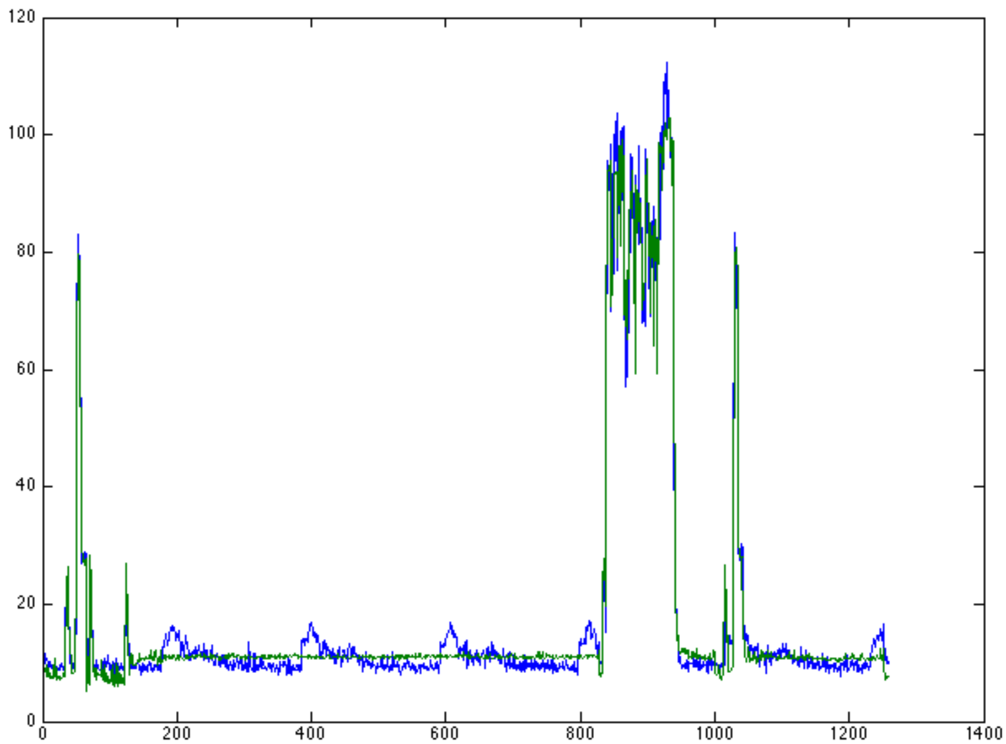


FIGURE 14: REGRESSION FIT FOR GPU

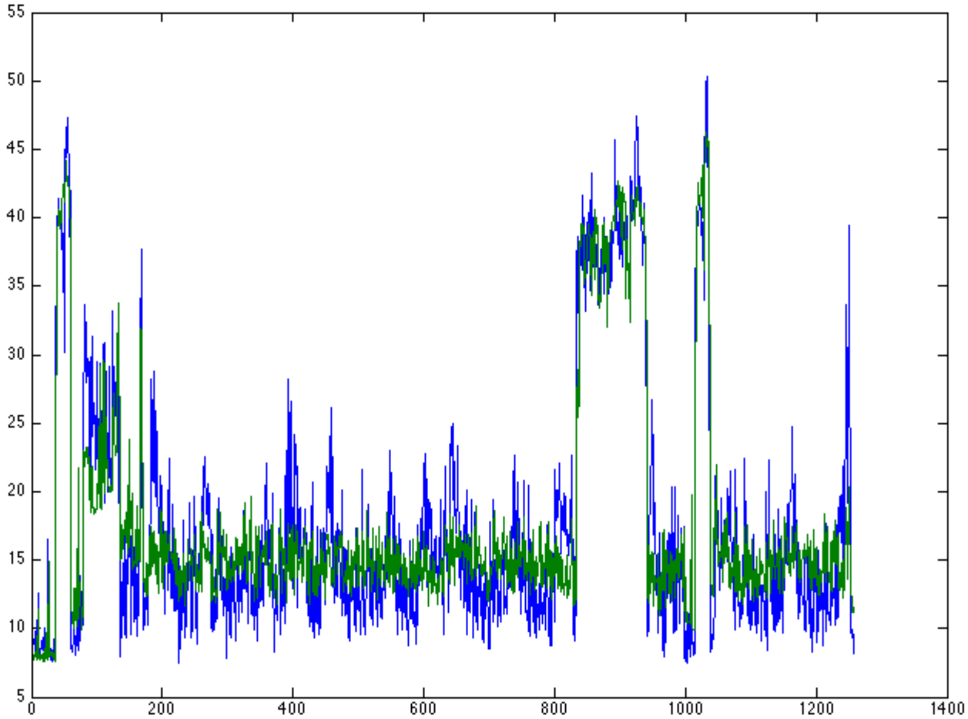


FIGURE 15: REGRESSION FIT FOR CPU

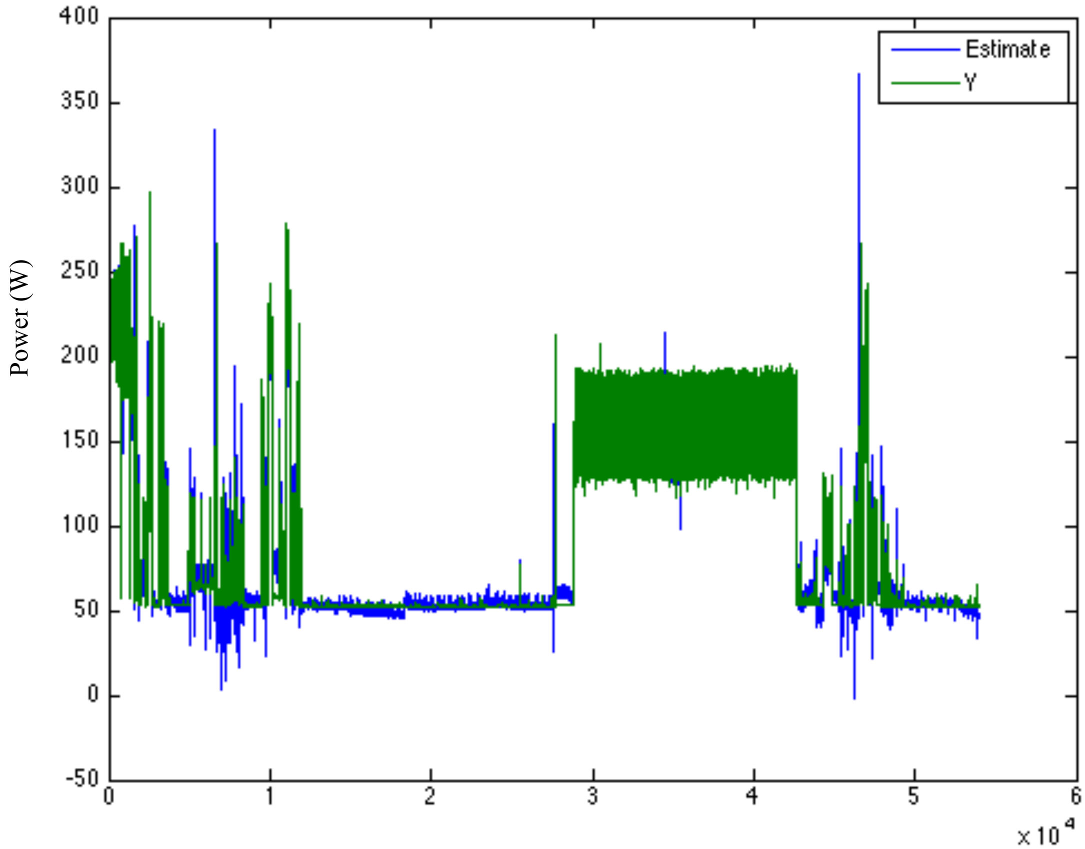


FIGURE 16: REGRESSION OVER THE ENTIRE STUDY FOR PC

Test on regression model revealed that regression analysis will not produce results with fidelity, because in one half of the graph the results showed lower than expected and another higher than expected. (Figure 17)

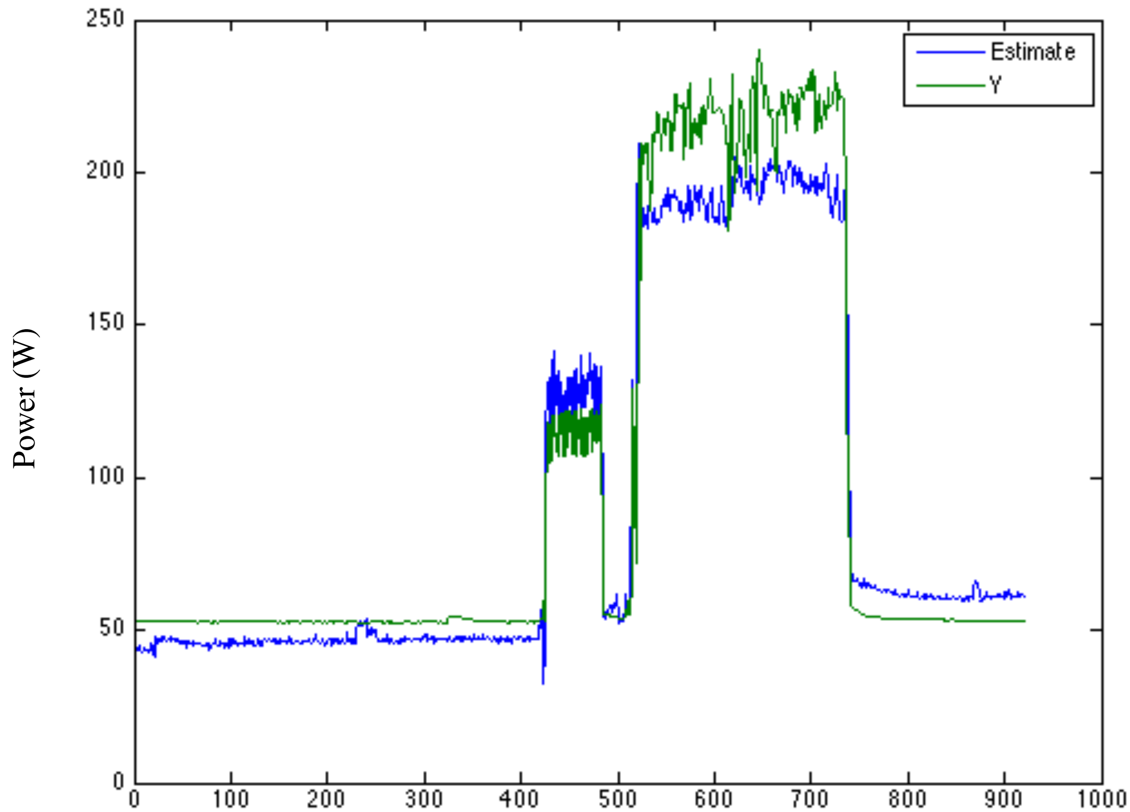


FIGURE 17: TEST ON REGRESSION MODEL

Artificial Neural Networks Method

Performance metric used was (MSE) Mean Square Error of data in training Validation and testing sets. The results converged in the 371st epoch using scaled conjugate back-propagation algorithm. (Figure 18)

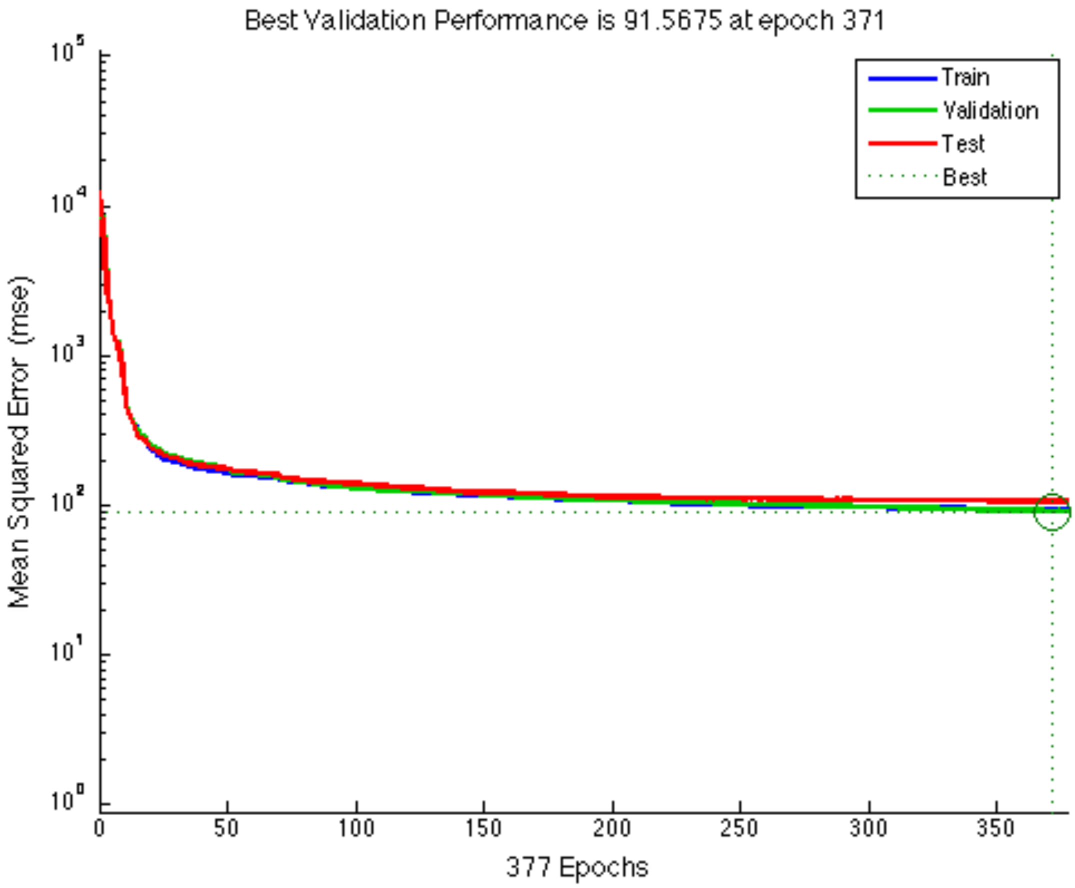


FIGURE 18: VALIDATION, TESTING AND LEARNING COST PER EPOCH

Regression of target and output for Training, Validation, Testing, and All Data sets

The results of artificial neural networks based learning are summarized in Figure below, where additional fitting lines represent goodness of fit.

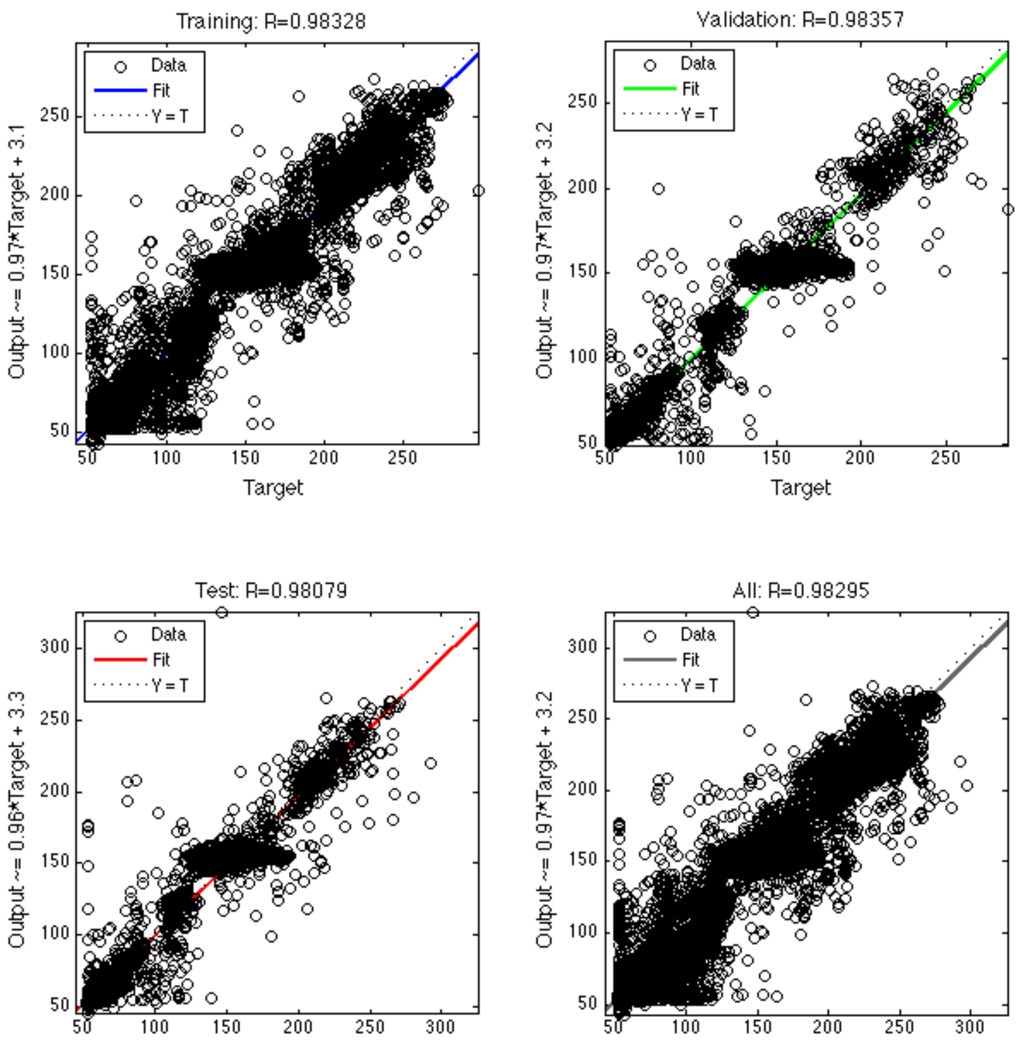


FIGURE 19: TRAINING, TESTING, VALIDATION AND ALL DATA ESTIMATES WITH RESPECT TO MEASURED DATA REPRESENTATION WITH REGRESSION LINES

Error Histogram.

In Figure histogram on artificial neural networks demonstrates errors, which are centered near 0.2 Watts. Distribution is symmetric and fits expected range of error.

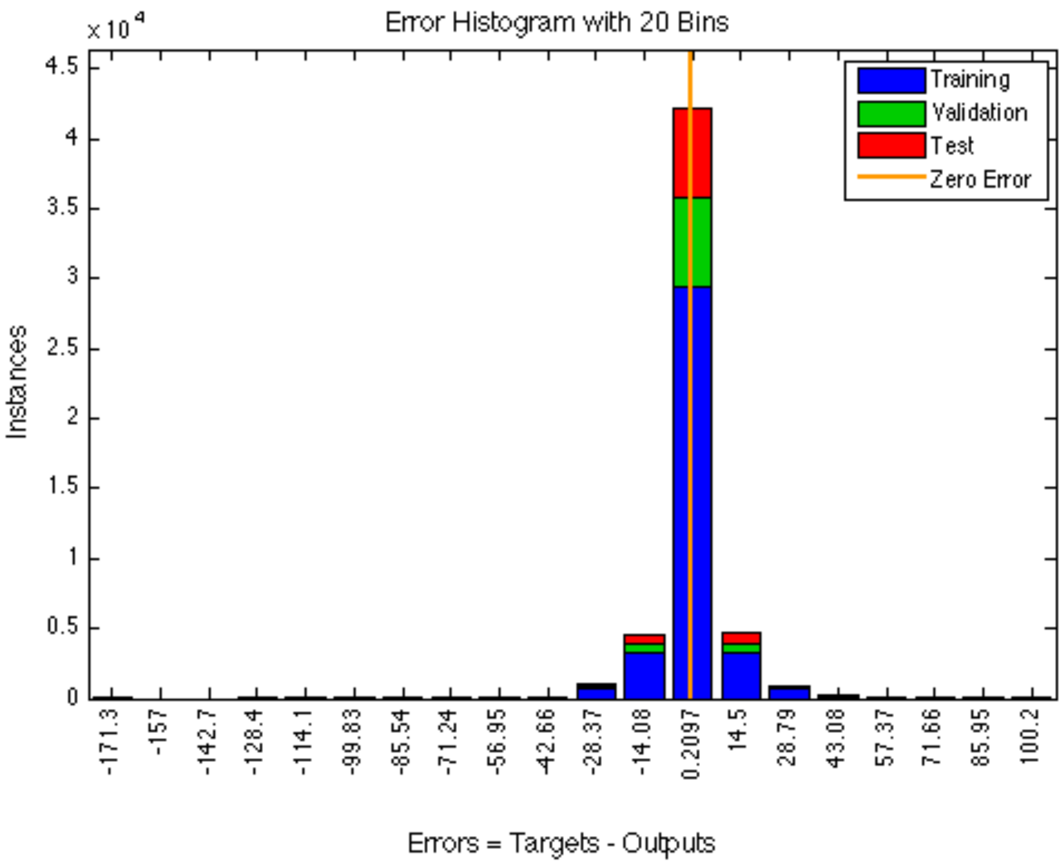


FIGURE 20: ERROR HISTOGRAM FOR NEURAL NETWORKS MODEL

Learned Model plot for output and target average power measurements per 2 seconds of over 50K data samples, demonstrates efficacy of the algorithm.

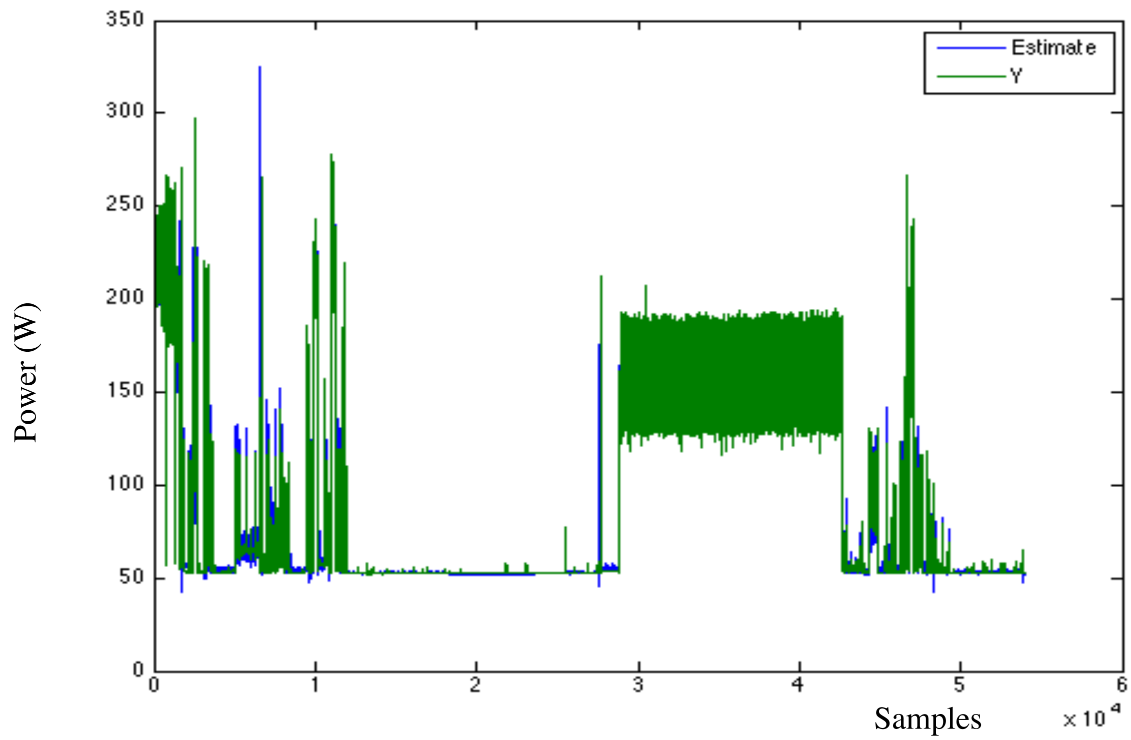


FIGURE 21: OVERALL PLOT OF ESTIMATED IN BLUE AND MEASURED IN GREEN FOR OVER 50K DATA SAMPLES

Test data, which was not incorporated with learning data sets (training, validation and testing of neural networks classification).

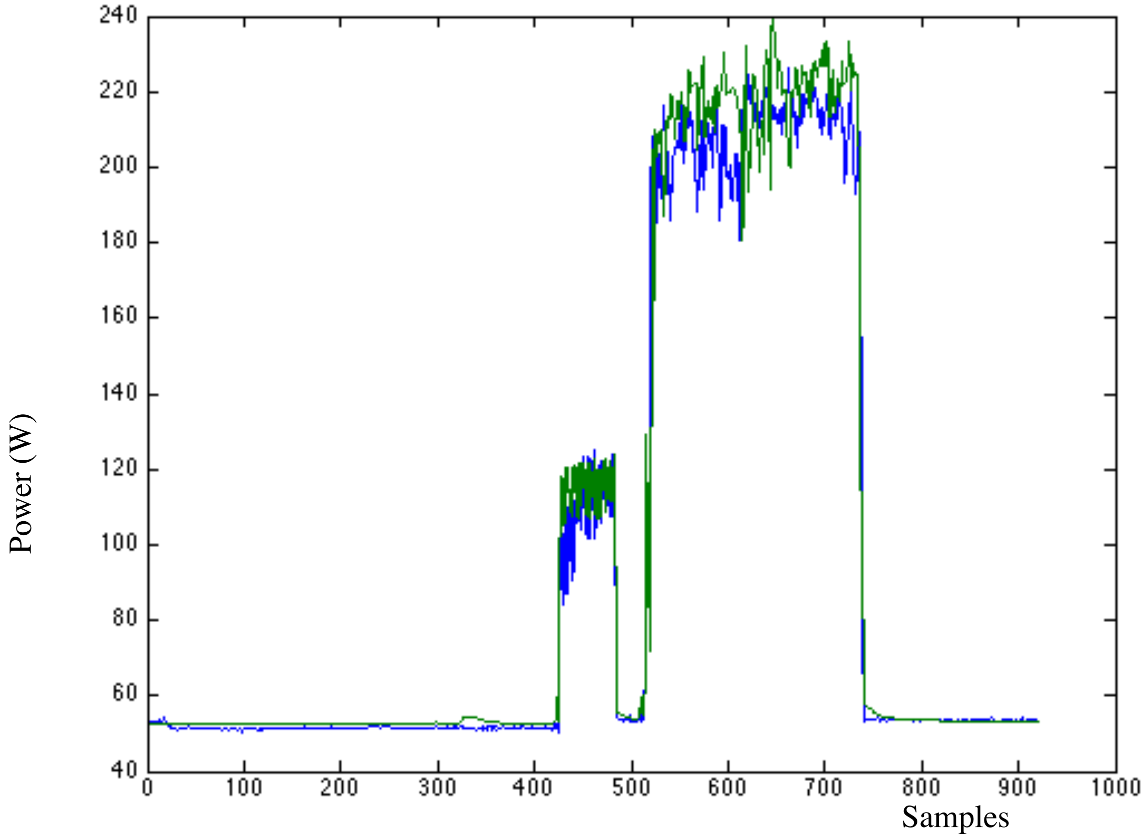


FIGURE 22: TEST DATA NOT USED IN MODEL FORMULATION AND REFINEMENT FOR PC

Error Histogram of the Test Data set in Figure Below, demonstrates with exception of some outliers results are centered.

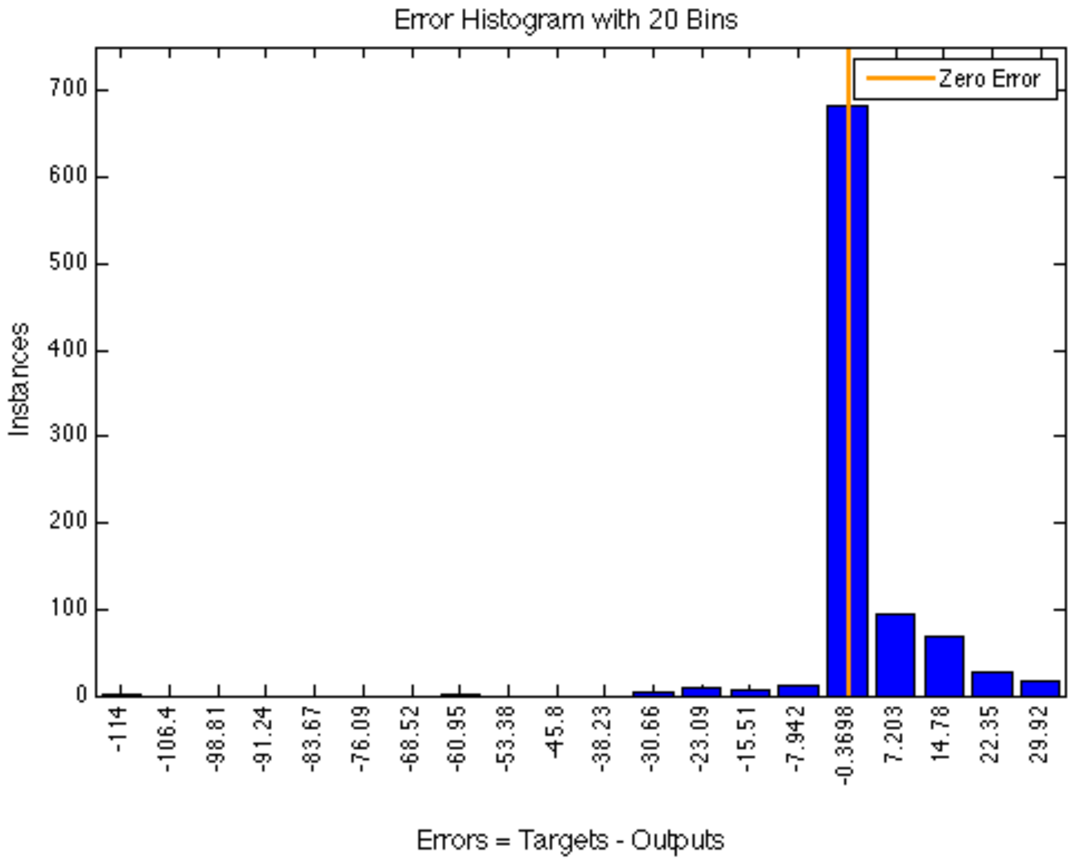


FIGURE 23: ERROR HISTOGRAM ON TEST DATA SET