MDPI

Dynamics of Research into Modeling the Power Consumption of Virtual Entities Used in the Telco Cloud

Etienne-Victor Depasquale [1,*] iD **, Franco Davoli** [2] iD **and Humaira Rajput** [3]

1   Department of Communications and Computer Engineering, University of Malta, MSD 2080 Msida, Malta
2   The National Laboratory of Smart and Secure Networks (S2N), Italian National Consortium for
     Telecommunications (CNIT), Department of Electrical, Electronic and Telecommunications Engineering and
     Naval Architecture (DITEN), University of Genoa, 16145 Genoa, Italy
3   Department of Electrical, Electronic and Telecommunications Engineering and Naval Architecture (DITEN),
     University of Genoa, 16145 Genoa, Italy
*   Correspondence: edepa@ieee.org

Abstract: This article is a graphical, analytical survey of the literature, over the period 2010–2020, on the measurement of power consumption and relevant power models of virtual entities as they apply to the telco cloud. We present a novel review method, that summarizes the *dynamics* as well as the results of the research. Our method lends insight into trends, research gaps, fallacies and pitfalls. Notably, we identify limitations of the widely used linear models and the progression towards Artificial Intelligence/Machine Learning techniques as a means of dealing with the seven major dimensions of variability: workload type; computer virtualization agents; system architecture and resources; concurrent, co-hosted virtualized entities; approaches towards the attribution of power consumption to virtual entities; frequency; and temperature.

1. Introduction

Several surveys of the results (see Section 1.6, "Related Surveys") of research into modeling the power consumption of virtual entities (VEs, i.e., virtual machines (VMs) or containers) have been written. In this work, our contribution lies in *a thorough analysis of the dynamics of research itself: the challenges, the approaches, the pitfalls, the fallacies, and the research gaps, without neglecting the fruits of the research*. Our intended audience is the prospective researcher, seeking to understand the dynamics of research into the predictive modeling and supporting measurements of power consumption by individual VEs relevant to *the telco cloud*. Dynamics are characterized through a thorough frequency analysis, which we conduct **through the application of a novel method we have developed** [1] that is unique in its ability to *parse* research literature. Through the visual aids we provide, and our observations through cross-cutting themes, a prospective researcher obtains a thorough characterization of the problems, approaches, developments, formal methods, pitfalls, fallacies and research gaps that characterize this research space.

Among the themes that our survey has brought us to identify, we have pointed out that all the problem categories we identified touch one or more of a set of seven main variables that may affect power consumption by virtual entities and the ensuing model representations: workload type, characteristics of the virtualization agent (VM or container), host machine resources and architecture, temperature, operating frequency, attribution of a fraction of consumed power to individual VEs and the mutual influence of concurrent VEs.

Among the major pitfalls that emerged from our thematic analysis, we highlight here the misconception of the Data Plane Development Kit's (DPDK) power efficiency (commonly misportrayed as a power hog), the often-unacknowledged limitations of the widely used linear models, the problematic use of benchmarks in model validation, the failure

to precisely identify the physical contexts of some experimental research, the influence of synthetic workload generators on measurements and the sometimes-overlooked relevance of processor organization on power consumption measurements. We have also pointed out the unavoidable need to precisely identify the scope and limitations of models and the fallacy of the quest for a "universal" power model.

The research gaps we identify are four: (a) modeling of containers' power consumption; (b) the effect of overcommitment on power efficiency; (c) investigation and classification of DPDK applications; and (d) modeling of power consumption by virtualized I/O (a challenge which is starting to receive some attention).

1.1. Why Is Research Needed?

Precise measurement of a VE's power consumption is difficult since measurements of its host's power consumption cannot be related directly to it. Hardware power meters are incapable of measuring the power consumption of individual VEs co-hosted on a physical machine. Moreover, power consumption of a VE varies with its hosting machine. Therefore, for VEs, accurate measurement is predicated upon precise *modeling* of energy- and/or power consumption.

A general prerequisite to devising energy- and/or power-efficient *operations* is accuracy in power and energy *measurements*. With specific regard to VEs, it is also essential for billing in multi-tenant environments, so that the Infrastructure Provider (IPr) can charge customers the fair amount for the resources (including energy) they consume. Before proceeding to the scope of our survey, we pose three salient questions that frame our work.

1.2. Why Should Energy Consumption Be a Topic of Interest?

Energy efficiency in the Internet (and in computing and telecommunication networks in general) has become a significant problem, which has received increasing attention since the early years around 2000 (see, e.g., [2–4] and references therein), starting from cloud computing infrastructures, and then extending to mobile and fixed networks. Indeed, it has been shown that the smaller data centers, within which telecommunications points of presence (PoPs) may be classified, represent around 95% of the United States' data center energy use [5]. Furthermore, this use is comparatively inefficient when compared with that of the hyperscale server farms (the remaining 5%).

1.3. What Is the Underlying Cause of Increased Energy Consumption?

Traffic growth is the primary cause of increased energy consumption. Table 1 shows the consistency with which Cisco's Visual Networking Index (VNI) has been predicting heavy growth in traffic exchanged over the access network by both businesses and consumers with:

- endpoints over managed networks;
- endpoints over unmanaged networks ("Internet traffic").

Table 1. Compound Annual Growth Rate (CAGR) reported in Cisco's VNI over four consecutive years.

Period	Fixed Internet Traffic	Managed IP Traffic	Mobile Data
2014–2019 [6]	23	13	57
2015–2020 [7]	21	11	53
2016–2021 [8]	23	13	46
2017–2022 [9]	26	11	46

(Note that the figures refer to compound annual growth rate (CAGR); they do ***not*** refer to the percentage share of total traffic.).

The key observation lies in the realization that, year after year, significant (heavy, in the case of mobile data) growth is persistently predicted. This observation is corroborated by several other researchers, with perspectives varying from traffic at the access segment

to traffic in transit between Internet Service Providers (ISPs) [10–12]. Cisco [6] and Sandvine [13,14] identify "video traffic" and "real-time entertainment" as the drivers of this growth. A later edition of Sandvine's Global Internet Phenomena report [15] dedicates its executive summary exclusively to video traffic; the report shows video as consuming 60% of downstream traffic—a further 2% increase over 2018. Alcatel-Lucent Bell Labs [11] observed that growth in the metro-core due to video traffic exceeds growth in video traffic crossing the long-haul core. Clearly, it is widely recognized that video is the prime driver of this growth in traffic and attendant energy consumption and research has been addressing this problem [16–28].

1.4. How Is Energy Consumption Being Tackled?

From these (referenced) works, the development of cache architectures emerges as an important approach to controlling energy consumption, but other mechanisms exist and yet more are emerging within the general thrust towards "future networks" (ITU-T Y.3001 [29]). Networks are evolving into flexible and programmable "softwarized" virtualized infrastructures, and strongly integrated paradigms, such as 5G [30,31], rewrite the research agenda. We may thus distinguish between radical and reformist approaches:

- Reformist approaches seek to improve the caching of content and are characterized by their investigation of the reduction of the length of the path between the source and destination of IP traffic.
- Radical approaches employ the dynamic and reactive control afforded by the softwarized, virtualized infrastructures. Radical methods are enabled by standardized architectures (e.g., [32,33]) that equip the control plane with uniform interfaces that exploit extant and emerging green capabilities.

Indeed, *the impact of virtualization technologies on power consumption in public telecommunication networks (PTNs) is still unclear*. There is a general belief that Network Functions Virtualization (NFV) should result in reduced energy consumption, owing to a consolidation of resources and increased flexibility in turning unused hardware (HW) on and off as needed. However, it is also true that "the massive introduction of general-purpose HW enabled by NFV would tend to increase power requests with respect to specialized HW solutions" [30]. Therefore, there is a need to operate power-aware management and control mechanisms in these environments. At the same time, it is necessary to limit the complexity of these mechanisms and the level of human intervention therein, to keep Operational Expenditures (OPEX) within reasonable limits. One approach to understanding this impact consists of comparative analyses of the implementations of infrastructure, with and without virtualization. This approach is taken in [34], where the evolved packet core (EPC) is studied. This work shows that the virtualized implementation is indeed less energy efficient. Unfortunately, the scope of virtualization and containerization within the converged wireless and wireline infrastructure is very broad and consideration of a single "use-case" [35] cannot be generalized to an overall statement. We therefore note that the scope for our survey needs an operational context which we suggest in the following.

1.5. The Scope of This Survey

1.5.1. Telco Cloud: The Operations Context

We suggest that the telco cloud is our network-operations context. "Telco cloud" is an evolving notion that evokes a number of common terms in attempts to describe it. Virtualization, software-defined networking (SDN), automation and orchestration are four such terms. Other prominent terms are edge computing, containerization, microservices and resilient infrastructure [36]. We suggest three key observations that organize these terms into a coherent image of the telco cloud.

1. **The telco cloud is, fundamentally, a hybrid cloud:**
   a. Self-sourced virtualization and containerization;
   b. Out-sourced (public cloud) containerization.

The complementary collaboration of the PTN operator's (PTNO) network, computing and storage infrastructure, with that of global providers of infrastructure and applications, is manifested well in [37]. A distributed cloud infrastructure operates at network (transport and interconnect) junctions. It includes (cloud) infrastructure owned and operated by the PTNO, by public cloud providers and by enterprises which consume their joint service.

2. **The telco cloud serves both internal and external clients** [36]:

    a. Internal use can suggestively be referred to as the ***IT Cloud*** [38]. This consists of applications specific to PTNOs: operational support systems (OSS) and business support systems (BSS), as well as more general applications, such as customer relationship management (CRM);

    b. External uses are growing organically on the basis of use cases seeded by ETSI [35] and the 5GPPP [39].

3. **The service-based architecture (SBA) of the 5G Core is a good fit with cloud-native computing**. Containerization is distinctively central to cloud-native computing [40,41]. The Cloud Native Computing Foundation explicitly identifies containers as components of the approach to the concept of cloud native computing [42]. There is a clear drive towards the use of containers in lieu of virtual machines as the operating environment for network functions [40], and the 5G Core's SBA provides a clear scope for employing containers.

We end this subsection by indicating the real estate where the VEs in our context may be deployed:

1. **Datacenters**: here, the real estate referred to consists of points of presence (PoPs) such as metro-core PoPs at the near edge;

2. **Softwarized and virtualized networks**: here, we refer to points of presence such as central offices (COs) and sites even deeper into the edge, such as remote radio head (RRH) sites and roadside cabinets.

1.5.2. Identifying the Models in the Scope of This Study

We survey **predictive** energy and power **models**, as well as **measurements** that facilitate the qualitative and/or quantitative prediction, of consumption by individual VEs relevant to the telco cloud. A simple interpretation of the rationale that drove our selection is that we have sought works that **measure real-time power consumption by VEs** and/or model **real-time power consumption by VEs**. The object of measurements and modeling is strictly the VE.

Nonetheless, the devil is in the details and therefore the details of this simple rationale must be worked out. One important, finer point regards the VEs themselves. There are software technologies, which we shall elaborate upon in later sections, which are **functionally critical** to VEs. Works that measure, and/or model, such technologies' power consumption are in scope. A justification of this claim on scope is not hard. Since power consumption is a scalar quantity, the reduction of power consumption of a component of a VE translates into the reduction of power consumption by the VE. In fairness, the translation is not direct (1:1). A generalization of Amdahl's law comes to mind: improvement in a component, measurable by some metric, is attenuated by the ratio of that component's use (measurable by that metric) to the system's (the VE's) use (measurable by the same metric). However, we can safely summarize the finer point we alluded to at the start of this paragraph as follows. Research that studies the measurement and modeling of power consumption by a component of a VE is in scope. We clarify this by example. For instance, we would:

1. **Include** the Data Plane Development Kit (DPDK) [43], as it serves the critical function of networking (VEs that serve as virtual network functions (VNFs));

2. **Include** a comparative study that measures power consumption by a VE using two different implementations of input/output virtualization technology, say: SR-IOV (single-root IO virtualization) and paravirtualization;

3. **Exclude** a comparative study that measures power consumption by various network adapter (or network interface card (NIC)) architectures, **unless** it reveals the impact of these architectures on VEs' power consumption.

Further detail emerges from the "real-time" requirement. This term is a reflection of the need for **application agnosticism**. Power meters that follow from such measurements and models may be used regardless of whether instantaneous or statistical readings of power are required. Studies (on measurements and models) meet this requirement by satisfying the following criteria.

1. Predictors:

    a. Must be of fine temporal granularity;
    b. Must be updated with the regularity of the temporal granularity;
    c. Must enable prediction of power consumption at the same temporal granularity.

2. Workload: Only resource-specific constraints are considered. That is, in the course of testing using, say, workloads that are processor-intensive (hence the workload is specific to the processing resource), no other constraints are allowed in works included in this study. In particular, models must not constrain the stochasticity of the workload.

Our last detail regards workload. We observe that the universal power model is a fallacy and the principal reason for this is that the interaction between workload and architecture cannot be pinned down indefinitely. This does not mean that modeling is a fruitless endeavor. It simply means that validity constraints must be placed on the model in terms of workload and architecture. Therefore, we do not exclude modeling and measurement because of its workload-scope or architectural scope. We do, however, observe that such models are pitfalls for those who apply them without knowledge of such limits.

We now proceed to present some recent, related surveys, highlighting their methodology and analytical approach. Against this background, we summarize the novelty of this survey.

1.6. Related Surveys

In recent years, much research has modeled the power consumption of servers in data centers and cloud environments in general. Several surveys in this regard cover different aspects of modeling and energy efficiency approaches for servers and virtual entities.

An analysis of power models the from micro- to macro-level is presented in [44]. This survey covers different aspects and levels of both hardware- and software-centric modeling techniques. Researchers studied models based on computing resources (CPU, memory, storage and I/O), system architecture (such as single or multiple cores), the availability of Graphics Processing Units (GPUs), system/network components, operating systems and virtualization environments. They categorize the existing models at different layers moving from architecture level modeling to the level of power models for whole data centers. However, the survey did not focus greatly on power modeling techniques that consider the effect of different virtual entities.

According to the categorization in [44], power models depend on different organizational contexts, including the power consumed by system components, running applications, and/or the execution strategy of processes. These metrics, however, can derive additive component-based, regression-based or machine-learning-based power models. Additive models usually present an aggregated view of server power consumption, which could be based on different resources (such as CPU, ram, I/O, disk), or the disaggregate static and dynamic power consumption of the server. Regression-based models are mostly based on the relation of power to the dynamic evolution of some measured system parameters. Power modeling using machine learning techniques is an advanced research area which can be further classified as supervised, unsupervised, reinforcement and evolutionary learning. Furthermore, at a much higher level, such as that of data center environments

as a whole, power models for groups of servers, datacenter networks, conditioning and cooling systems were studied. The survey also analyzed the modeling methods at operating system (OS) and virtualization level, and for data-intensive, communication-intensive and general applications. At the end, the researchers compare the power models against their complexity, effectiveness, application and use-cases.

The survey carried out in [45] analyzes the power models based on their modeling approaches. It claims that power modeling methods can be divided into two main themes: analytical models and formula-learned models. In the former, parameters affecting the system power are known to the researcher; however, the weight for each parameter needs to be determined. On the contrary, in high-level formula-learned or machine-learning-based modeling, no prior knowledge of the system is required and the model is developed from scratch using the provided data set. The survey further describes commonly used machine learning techniques, and later focuses on the use and effectiveness of neural networks in power modeling. According to this survey, the approach to model power for any server may vary in several aspects. These different characteristics are summarized as (i) *degree of autonomy*, i.e., to which level the power model is dependent on external hardware; (ii) *level of granularity*, which describes the depth of logical (core, thread, etc.) or physical (device specific, system level, etc.) levels to which the model can precisely estimate the power; (iii) *methodology*, describing the selection of method, which could be simulation, analytical modeling or data-training; (iv) *simplicity*, which can be assessed by the number of variables, method selection and models' computing overheads; (v) *portability*, as most of the models developed are effective for some specific architecture, workload or environment, and their generalization is still a question; (vi) *accuracy*, which defines the measurement accuracy or estimation precision of power for any model; and (vii) *power meter* (as most studies use external power meters as a ground truth for their modeling, their accuracy is also a major concern for developing a more efficient model).

The survey in [46] adopts a slightly different analytical approach and evaluates selected existing power modeling techniques in a unified environment. Comparative analysis has been performed for twenty-four different software power models and measurement methods, with nine different benchmarks under a single experimental environment. It evaluates the existing software power measurement techniques and models for different applications, benchmarks, systems configuration, server architecture and for their estimation errors. The authors claim that most of the software-based power models use system performance metrics provided by the operating system, or performance monitoring counters provided by hardware sub-systems of the server. Software power models considered in this study are categorized in three types as single variable CPU-based, multi-variable CPU-based, and single-variable throughput-based. The result of this unified experimental setup shows that power models based on a support vector machine (SVM) and interpolation techniques show the least error for different resource-intensive applications, whereas lasso regression with 30 variables was found to be the worst power model with the highest error. Furthermore, the modeling techniques are mainly divided into two categories; linear and non-linear, where each category is further classified based on its derivation from mathematical modeling or machine learning techniques.

Another survey [47], with a more limited scope, focuses on the power modeling of servers in the cloud while considering the complexity stemming from the diversity of host hardware platforms, virtualization environment and workload. It presents the usability, applicability and limitations of the studied approaches, and also presents the analysis of the traditional existing and emerging modeling techniques. It reviews the existing power modeling methods at three stages: data acquisition methods, power consumption models and power modeling methods for servers, VMs and containers. According to this study, the collection of data for power models could be based on: (a) instruments such as external power meters, (b) dedicated acquisition systems which are generally products customized by developers for a specific hardware, (c) simulation-based, and (d) software monitoring tools. The latter constitute a widely used method that is based

on system indicators and sensors. For the second stage—i.e., the power modeling—either hardware-centric, virtualization-centric or application-centric power modeling schemes can be used. Each type has its own drawback and usability; hence, the domain is still an open research area for further generalization and measurement precision. The selection of method, which is considered as the third step of the modeling in [47], is characterized as power-modeling-based on: (a) empirical parameterization, (b) function regression, (c) machine learning, and (d) evolutionary algorithms. The survey, after analyzing different power models, concluded that functional regression and machine learning methods yield high accuracy when provided with large enough data sets and clear power behavior of servers. This survey is more similar to ours; however, we distinguish further by providing statistical analysis of existing approaches, used tools, methods and measurement metrics. Moreover, our survey focuses on the modeling of power for individual virtual entities and on the effect of the virtualization environment on server and network functions' power behavior.

In general, the surveys mentioned above present the literature on server models irrespective of the operational environment. One of the major trends, not only in data centers but also in networking, is the presence of virtual entities and the adoption of virtualization environments. In contrast, our survey concentrates on the power models of virtualized entities, and focuses mainly on components and parameters that could affect their power consumption. It further provides a statistical analysis of included research works with respect to their data acquisition methods, measurement tools and modeling methods, to give researchers a bird's-eye view of the existing literature through which one can identify the approaches carried out in different studies.

1.7. Organization of This Survey

The survey is organized as follows:

1. In [1], we describe our method: a problem–approach–development (PAD) triad, which, to the best of our knowledge, we are the first to use to identify research dynamics. We have delegated the description of the method to a separate paper, to resolve the difficulty of elaborating fully on the method without distracting attention from the results which we have obtained and which we document separately in this review.
2. Section 2 presents the detailed results.
3. Section 3 is our analysis of the results. There, we give a qualitative assessment through themes which emerged as we organized the data. We have classified these themes as "state of the art", "fallacies" and "pitfalls", to suggest guidance and warnings which we were able to glean from others' experiences.
4. Section 4 concludes by attempting to encapsulate the insights we have gained through this work.
5. Appendix A illustrates the use of structural coding on a sample of the corpus on a popular research area: predictive models of renewable energy consumption in the radio access network.
6. Abbreviations lists and expands the acronyms used in this paper.

We complement this paper with an online repository (https://github.com/humaira-salam/PowerMeasurementAndModelingRawData, accessed: 23 December 2022), that carries our raw data.

2. Survey Results: A Digest of Challenges, Approaches and Developments

2.1. A Taxonomy of the Problem Space

As our parsing of the literature proceeded (our method is described in [1]), we observed that the scope of this survey is relatively narrow and the problems in our set are not fully independent of one another. Rather, the problems diverge from one another only as *aspects* (we could also say that they are *derivatives*) *of the core challenge of modeling the power consumption of virtualized entities*. Each RU (research unit) (The RU, or unit of research, is "a publication (excluding surveys) in conference proceedings and

journals" that "ha[s] three common manifest properties", i.e., problem(s), approach(es) and development(s) [1]) is rooted in this core challenge, but the derivative problems (our *Problem categories* (P-categories) and their members) addressed differ from one RU to another. Figure 1 is an illustration of a simple organization of the challenges which have been tackled in the literature and shows their frequency of presence in RUs. The organization gives prominence to how challenges have been perceived:

1. One group regards the concern with obtaining an understanding of the dependency of power consumption on some genre of artifacts. Categories P1,2 and P9–11 are in this group;

2. The other group regards the concern with how to predict power consumption. Categories P3–8 and P12 are in this group.
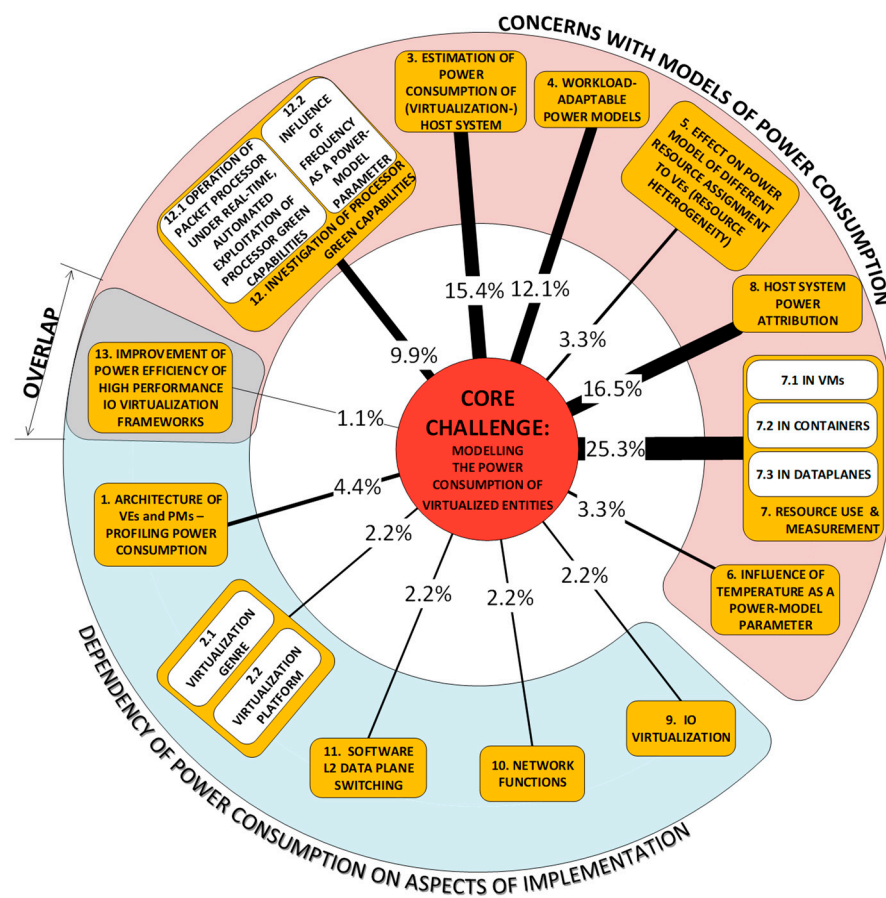

Figure 1. The core challenge and its derivatives; research interest R_{P_k} is shown in line thickness and as percentage.

We now proceed to describe the categories in more detail. Each description is preceded by a list of references to works that tackle a challenge in the problem category.

2.1.1. Problem Category P1: Host System Hardware Architecture Perspective: Dependency of VE Power Consumption on Host System Architecture [48–51]

Challenges in problem category P1 address the impact of specific architectural attributes of the host system on the power consumption of VEs. They relate to changes in power consumption (the behavior) as major attributes of architecture and system-level designs are adjusted, inserted or removed. Insertions and removals are coarse configurative actions such as enabling or disabling; adjustments consist of progressive modifications such as adding increments of a resource. Examples of attributes which have been tackled include multiple processor cores, processor frequency scaling, Non-Uniform Memory Access (NUMA) and hardware threads (e.g., Intel Hyper-Threading). For network functions,

the importance of knowledge about the power efficiency of NUMA and multiple-core architectures has the added relevance of these architectures' relationship to ***determinism*** [37]. We dwell further on the underlying premise of ***hard partitioning*** in our consideration of the impact of the high-performance data plane on power efficiency (see Section 3.3.2).

Research that investigates the dependency of power consumption on architecture is ***exploratory, charting*** work. It attempts to provide a framework for detailed modeling through the discovery of broad relationships. Problems in this category arise with developments in architecture and system-level design. For example, while [49] is a comparatively old work that tackles architecture, [50] is newer and finds scope for research in system software's exploitation of NUMA. One recent, highly significant scope is that of the use of ***domain-specific architectures (DSAs)***. Researchers are exploring specialized hardware in the quest for the improvement of the energy–performance–cost ratio, and will investigate energy efficiency in the process of their research. As domain-specific cores are mixed with general-purpose cores, many architectures will be investigated from each of the three pinnings: energy consumption, performance and cost. A particularly relevant set of DSAs regards real-time packet processing by computer systems hosting NFs at intermediate nodes (INs) at the network edge. Concern lies with expediting the common tasks, such as sending/receiving packets and processing headers. SR-IOV is a good example (see, for example [52], and its inclusion in [48]), but software-only solutions, such as poll-mode drivers, may also help to cut through the many middlemen characteristics of general-purpose operating systems [48,53]. Introduced to serve the perspective of performance, it is now necessary to understand their impact on power efficiency. Therefrom, it is necessary to understand how to ***control*** their power consumption. We suggest that "***profiling***", the term chosen in [49,54], is a helpful descriptor of this kind of research. Just as a profile produces an external boundary within which to fill detail, so does this kind of research provide a framework through which modeling work is facilitated and within which modeling work provides details of power consumption relationships.

2.1.2. P2: Impact of Alternative Virtualization Genres and Virtualization Platforms on VE Power Consumption [55,56]

Here, a behavior that consumes power is investigated across different ***implementations*** of a ***system concept***. We have observed investigation of two different system concepts: (a) virtualization genres and (b) virtualization platforms. There are three members of the virtualization genre group: containers, para-virtualization and hardware-assisted virtualization. In the virtualization platform group, examples include Xen, Hyper-V, Kernel Virtual Machine (KVM), Docker and Linux Containers (LXC). Research questions typical of category P2 seek to control the scope of experimentation through exercise of specific resources, e.g., per-host networking using emulated switches (software switches) [55,56], processor-bound and memory-intensive processing [54].

We consider genres and platforms as sub-categories of the same overarching problem category. Namely, this is system-level exploration that attempts to establish generalizations about an uncharted space. Like problems in category P1, new problems in this category arise with fresh alternative virtualization genres and platforms. However, here the scope of investigation is broader than with works classified under P1. Unlike P1, where specific architectural aspects (e.g., NUMA, hardware threads) are explored, the perspective taken here is a concern with the impact of the choice of implementation of a system.

2.1.3. P3: Estimation of Power Consumption of (Virtualization-) Host System [57–69]

Measurement of a single server's power consumption through the use of an external power meter is a trivial task. However, at the scale of cloud datacenters, it is a logistical burden. In addition, travel to the datacenter's site may be burdensome. Furthermore, service availability would be reduced by the process of attaching a physical power meter to the hardware in the virtualization platform, e.g., between the server's power inlet and the outlet in the racking cabinet's power distribution unit (PDU) (naturally, availability would

only be affected in cases that do not integrate (management and) measurement facilities within the PDU).

The alternative is the deployment of software power meters. In the scope of this survey, the cases we consider are meters that attempt to predict host power consumption on *the basis of activity in the VE*. This challenge is tackled, for example, in [58–60,62,67,70]. These works then proceed to tackle the problem of attribution of system power to the guest VEs. Indeed, inclusion within the scope of both challenges (modeling power consumption of VEs and that of the host system power) seems to significantly enhance the usefulness of such research, with relatively less effort.

Host power consumption may be predicted in terms of VE resource utilization, or in terms of *a simple characterization of the VEs' workload*. The use of simple workload characterization as a predictor requires knowledge of workload parameters such as the number of processes, number of threads, web interactions per second and network interface utilization. Enokido's and Takizawa's work [67,68,71] is noteworthy in its consistency in modeling in these terms but other variants of this approach have been found: (a) web interactions per second [65] and (b) number of VMs running processor- and/or network-intensive workloads [66].

To contrast: works such as [57,72–76] are not included within this category, notwithstanding their development of models for the prediction of host system power consumption. In these works, host system models were developed as part of the scope of the challenge of modeling virtualized entities. Therefrom, the challenge of system power attribution (problem category P8) was tackled to proceed to guests' power models.

2.1.4. P4: Dependency of Power Model on Workload [56,62,67–69,71,75–79]

This category regards the perceived dependency of a VE's power consumption model on the tasks it is processing. While it is intuitive to expect power consumption to depend on the workload, it seems far less intuitive to expect the model to depend on the workload. If this dependency is detected, the problem of model formation must undertake this aspect of investigation. Two different, major approaches towards achieving *adaptability* of the model to the workload have been observed:

1. *Adaptation during run-time:* the selected mode of instrumentation may not be suited to a generalizable, closed-form relationship between inputs and power consumption. In this case, model parameters must be re-trained online. This approach is therefore of the operating-time, or run-time, kind;
2. *Off-line adaptation:* a larger set of inputs may need to be identified to comprehensively characterize the variation of power consumption with the workload. This approach is therefore of the design-time, or off-line, kind.

We conclude this part with a note about two descriptors of the workload: homogeneous and specific. The term "homogeneous" is encountered in the literature to refer to the case where host system deployments within scope are subjected to a single workload. The term seems to originate in warehouse scale computing (WSC). Conclusions drawn from this kind of workload have drawn criticism as the results, while significant by virtue of the mass of WSC, are not generalizable. The other term—"specific"—identifies a single application; for example, a member of the Standard Performance Evaluation Corporation (SPEC) CPU2006 suite [80]. This term is used to indicate that models tested under such a workload are application-dependent and are valid only within a limited range of this dimension of variability (i.e., the "workload" dimension, see the treatment of the seven dimensions of variability).

2.1.5. P5: Dependency of VE's Power Consumption and Power Model on VE's Resource Configuration (Heterogeneity) [62,73,78]

This category regards the perceived dependency of a VE's power consumption and/or the dependency of its power consumption model on (a) the physical host's resource configuration and (b) the individual VE's resource allocation. Research here is concerned

with two cases of very practical problems: the impact on power consumption of (a) the differences between hosting machines/containers and (b) the differences between virtual machines. We have observed that occurrences of research that undertake this challenge tackle it as an adjoint to another focus, not as the research's primary objective.

1. *Physical host configuration:* Host machines in a cloud datacenter may be expected to come in a limited variety of types, principally differing in resource capacities such as the number of processor packages per server, cores per processor package, amount of RAM per server, spread in storage device sizes, etc. Processor power consumption is notably variable, even within a single family of processors. Indeed, specialization in optimized power consumption within a family of processors is a part of the study carried out in [34] within the context of an edge cluster for use in NFV. As a VE migrates from one processor within a family to a processor of a different specialization, its power model will change.

2. *Individual VE's resource allocation:* The power consumed by a VE varies with the allocation of resources to (i.e., in use by) a VE, which can be dynamically varied. The number of virtual cores assigned to a VE is a notable example, see, e.g., [57,61,72–74]. Moreover, VEs are commonly offered in sizes, e.g., small, medium and large, where allocation varies within all the major resource categories, demanding prediction of power consumption matched to the size of the purchased VE.

2.1.6. P6: Impact of Temperature and/or Frequency on Models That Predict VEs' Power Consumption [60,81,82]

This category regards the challenge of the inclusion of processor package temperature in models of power consumption. Works that tackle this challenge are concerned with detailed models of power consumption. Here, the interest lies in obtaining models that incorporate dependence on hyper-parametric attributes such as temperature.

2.1.7. P7: Loading the VE's Resources and Measuring Resource Use [56–59,61,62,65,68,69,71,73,75–79,81–87]

This category regards the use of computing resources and the measurement of such use by VEs. Interest stems from the role of resources *as predictors in modeling*. The researcher is firstly concerned with *loading* (i.e., effecting the use of) resources. What means *within the operating context of the VE* can be used to load a resource? Should it be loaded in isolation (using synthetic loads) or should it be loaded using representative (realistic) workloads? Once these problems have been addressed, the concern with the *measurement* of resource use arises. The problem here consists of identifying the means that quantify resource use made by the loading.

2.1.8. P8: Attribution of Host System Power Consumption to Individual VEs [57–59,61,62,73,75–77,79,81,82,84,86]

The attribution of host system power to individual VEs is a fundamental problem in proceeding from the directly measurable (host system power consumption) to the indirectly measurable (individual VEs' power consumption). Direct measurement of host system power is possible (e.g., at the wall outlet, or through voltage rail in-line resistors), and such empirical evidence can be used as a ground truth and compared with power consumption inferred through modeling. How, then (and herein lies the problem), can this consumption be attributed to the host's individual guests (the VEs)?

Within the host system, power consumption may be divided into *idle (static), active (dynamic)* and *overhead*.

1. Idle (static) power consumption:

    a. Power consumption while idle is not attributable to any VE at all, as this consumption arises out of the electronic behavior of semiconductor material, not of computation, communication or storage;

  b. Nonetheless, this power consumption must be accounted for and different approaches have been followed. For example, the physical machine's idle power is attributed to individual VEs in fractions equal to the ratio of each VE's virtual CPUs (vCPUs) count to the total complement of vCPUs active on the physical machine [57,72–74].

2. Active (dynamic) power consumption:

  a. The active component can be linked to a particular VE;

  b. This includes active power consumption in peripherals, e.g., network interface cards/adapters (NICs) and mass storage devices.

3. Overhead, e.g.:

  a. Operation of heat dissipating units (fans) to prevent thermal runaway;

  b. Losses in the power supply.

A "top" (host)—"down" (guest) approach to attribution has been observed.

1. Decide on what host system power consumption is within the scope of the study and how to divide it. The problem of attribution of the above three causes may be summarized as follows:

  a. Is idle power attributed to the VEs or is it attributed to the host/a privileged guest?

  b. Is consumption by peripherals within the scope? How will this be attributed?

  c. Are overheads modeled or is correlation with other sources of power consumption going to account for them?

2. Select a set of performance metrics that are correlated to a VE's power consumption.

3. Select a model that maps a VE's performance metrics to its power consumption.

4. This fourth consideration is tackled only by those researchers who investigate the *adaptability* of the attribution obtained through steps 1, 2 and 3. Does the obtained attribution adapt well to concurrent, co-hosted VEs? That is: if concurrent, co-hosted VEs were to be investigated, would the division, metrics and model still result in accurate prediction?

2.1.9. P9: Implementation of Virtual I/O; P10: Implementation of Network Functions; P11: Implementation of Software Layer 2 (L2) Data Plane Switching [48,51,53]

These three categories are introduced together, since elements from the respective categories are commonly implemented *as a set* for the purpose of the realization of the virtualization of network functions. Here, researchers seek comparative statements and/or broad correlations (e.g., independent, positive, negative) between the workload (often in terms of packet rate and size) and power consumption, across implementations of the same type. As was observed for categories P1 and P2, researchers seek a profile of the power characteristics of implementations. It may be helpful to repeat that by "profile", we understand that characteristics sought here are not of the detailed form of closed-form expressions. Examples of elements from the respective categories are:

1. Virtual I/O (P10): virtio [88] and DPDK poll-mode drivers (PMDs) [89];

2. Network functions (P11): Bro (now Zeek) [90] and Snort [91];

3. Software layer 2 data plane switching (P12): Open Virtual Switch (OvS) [92] and VALE [93].

Problems in each of these three categories merit separate classification as they have been tackled separately in the literature. For example, in [53], a number of components are investigated: three different implementations of software virtual switch (P12), two different I/O virtualization devices and two different implementations of the same network function (intrusion detection system (IDS)). In [51], power consumption by packet transmission under DPDK is investigated under the condition of enforcement of (a) the network adapter's affinity to NUMA nodes and (b) DPDK process pinning to processor cores.

2.1.10. P12: Investigation of Processor Green Capabilities [50,60,67,81–83,85,87,94]

Works in this category investigate the low-power idle (LPI) and adaptive rate (AR) operation of a processor as a means of reducing power consumption. The challenge is broad enough to permit a sub-categorization into (a) those works that investigate the influence of frequency as a power-model parameter [81,82] and (b) other works that address improved, real-time governance of LPI and/or AR [83,87,94] to minimize the power consumed to process a load.

2.1.11. P13: Improvement of Power Efficiency of High-Performance IO Virtualization Frameworks [87]

A separate classification was set up for [87] as this work represents an evolution of those classified under P9. This work extends beyond profiling and suggests use of low-power instructions as the means to balance performance and power efficiency.

2.2. A Taxonomy of Approaches

Figure 2 illustrates the taxonomy we use to structure the approaches detected in research work. Line thickness and percentage values represent the *utility* of the specific approach. Utility is best understood within the context of all the observed triads in a literature corpus. For any specific approach, this may be used to solve a variety of problems and its application may result in a variety of developments. One may therefore think of the approach as a *nexus*, or a point of confluence through which many researchers pass as they attempt to solve problems. Thereafter, researchers diverge radially outward from this point of confluence towards some achievement (some development). A suggestive image is to think of the approach as the center of a star, but spokes converge onto it from problems and diverge away from it onto developments. When the count of these triads (each composed of two radial lines, or dyads) is divided by the sum of all such counts for all approaches, we obtain a metric: a normalized quantity obtained within the context of all approaches. See our method [1] for the formal statement of *utility*.

We now proceed to describe the categories within the context of the taxonomy. Each approach-category's description is preceded by a list of references to works, each one of which uses a component within that category's set.

2.2.1. Analytical Foundations

This group of categories regards the theory and hypotheses that comprise the essential abstractions at the basis of scientific research.

A1: Power attribution principle [48,53,57–59,61,62,69,73,75–77,79,81,82,84,86]: When *host system* power is measured, whether at the wall outlet or at one or more of the power supply's output lines, there is the problem of attributing the measurement to the logical divisions (VEs) of the host computer system. The attribution of system power to modeled entities starts with a decision on which power consumption is within scope (see Section 2.1.8). Next, power in scope is attributed to (burdened on) one or more entity. For example: will idle power consumption be attributed to the host system or will it be attributed to the VEs?

A2: Modeling bias [57–64,72,75–77,79,81,83–85,87,94]: Researchers approach the problem of developing a model under some bias which conditions their final outcome. This bias is manifest in researchers' selection of a particular type of regression to apply to their data. We note that this same observation is carried in [95]. Here, our purpose is solely to draw attention to what we have observed as researchers' **modus operandi** without analyzing their choice of approach.
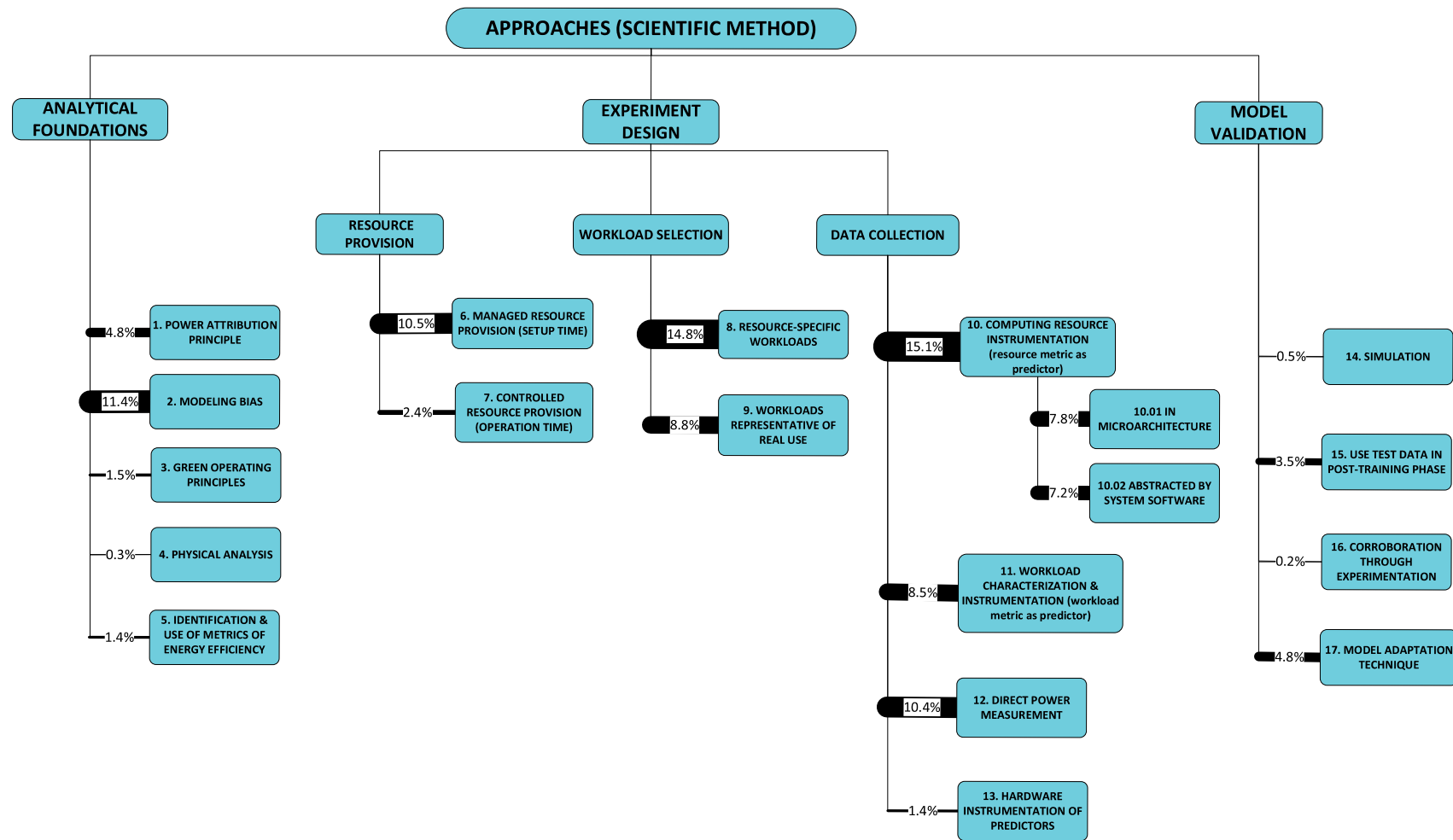
Figure 2. A taxonomy of approaches; approach utility U_{A_k} is shown in line thickness and as percentage.

A3: Green operating principles [83,87,94]: Works in this category weave radical approaches to power efficiency into their developments. For example, instead of conventional scheduling, ***run-to-completion*** [96] is exploited to obtain dedicated (or, at least, very sparsely shared) resources for the processing of packets. This approach is further nuanced by the real-time control of adaptive rates and sleep depth. In one particular case [87], the novel concept of a low-power instruction instead of transitions to/from low-power idle (sleep) states is used.

A4: Physical analysis [60]: This category regards approaches rooted in the physical properties of (semiconductor) material in the consumption of energy. Only one work [60] was found fitting this category. However, another two that used this approach to study the power consumption of physical entities (and therefore was outside the scope of this study, which is concerned with VEs) were found and they are described next to illustrate the approach better. In [70], a study implicitly applies Dennard's law in the process of obtaining weights that scale a processor sub-unit's contribution to power consumption. In [97], the physical cause of power consumption in metal-oxide-semiconductor (MOS) material is examined and used as the basis for modeling equations.

A5: Identification and use of metrics of energy efficiency [49,50,65,85]: The relationship between system architecture and power consumption can be investigated through the identification of the relevant metrics of energy efficiency. For example, an easily recognizable metric, albeit rather broad in possible interpretations, is the J/b (joule/bit). The use of such metrics encourages joint consideration of function and power consumption.

2.2.2. Experiment Design

The practical, hands-on aspects of the empirical process are the product of a (probably cyclical) design phase, concerned with a number of stewarding activities pertinent to test subjects and ancillary objects in the testing scenario, instrumentation, inputs and outputs. The activities include selection, configuration, interconnection, initiation, observation and termination. We have identified several examples of such activities within our research scope and grouped them under ***resource provision*** (categories A6, A7), ***workload selection*** (A8, A9) and ***data collection*** (A10–A13). We describe these categories next. Admittedly, the activities referred to (i.e., selection, configuration, etc.) have broad meanings; therefore, in the course of describing the categories, references to the activities are emphasized by bold, italicized text.

A6: Managed resource provision [48–51,53,55,56,60,66–68,73,81–83] (***selection, configuration***): This concerns the provision of resource capacity either to a specific VE (the guest system) or to the physical entity (the host system) hosting the VEs. Within the empirical process, the techniques in this category provide the means to observe the effect on the power consumption of managed changes in resource provision. Examples include the (manual) pre-***configuration*** of:

- The frequency of operation of processor cores [48,50,81];
- Core affinity [51,55] and hardware-thread affinity [68];
- The network interface data rate capacity capping [66].

These techniques are executed as part of the process of ***selection*** of the operating parameters, i.e., setting up experimentation, ***before*** operations start. This qualification is necessary to distinguish from such approaches as may change the operating, run-time context.

A7: Controlled resource provision [78,83,85,87] (***configuration***): Provision of resources may change during the running of an experiment (rather than before it starts). Approaches in this category include the automated adjustment of:

- Processor frequency (also known as performance state, or P-state) [78,83,85,87];
- The depth of processor sleep (also known as low-power idle state, or C-state) [83];
- The number of hardware threads [78];
- The time spent running a low-power instruction [87].

These techniques are approaches to solving the problem of full-throttle operation. Without a guided operation of adjustments such as those listed above, operation of the processor may quite reasonably be likened to a multi-assembly-line manufacturing plant that operates line machinery whether there are goods to produce or not.

A8: Resource-specific workload, A9: Representative workloads *(selection, configuration, interconnection)*: These two categories regard the *workload selection* stage, within experiment design in the scientific method. The workload comprises the *inputs* referred to earlier; inputs must be *interconnected* to the system under test, and this is often not a trivial task. In our thematic analysis in Section 3, we identify *workload type* as one of the seven dimensions of the variability of power models. The influence of workload type on the model obtained is evident in the attention paid by researchers to their selection of workload type. We can distinguish two broad categories of type.

Resource-specific workloads (A8) [48–51,53,55,56,59–61,63,64,66,67,69,71,75–77,84,86] are applied to investigate the impact of the utilization of specific resources on power consumption. Such *synthetic* workloads are applied *(interconnected)* to a machine (whether virtual or physical) to reduce (as much as possible) the scope of power-consuming resources to a targeted set. Resource-specific workloads are most commonly used in exploratory work, to gain an understanding of the behavior of a resource's power consumption. We refer to this approach as resource-specific workloads, synthetic workloads or resource isolation.

Representative workloads (A9) [48,56–58,60,62,65,68,69,73,75,78,79,81–85,87] may be used as complementary with, or alternative to, resource-specific (synthetic) workloads. A notable complementary use is made in the testing (post-training) phase of model development, when representative workloads are used to validate a model (obtained using synthetic workloads). They may also be used in an entirely alternative approach to synthetic loading, to support development of application-agnostic models. Representative workloads lead to training data that incorporates variation in utilization of more than one resource at a time; hence providing at least limited application agnosticism.

The next four categories (A10–A13) regard the *data collection* stage, within experiment design in the scientific method. Categories A8 and A9 regard the *selection* of workload type. The approaches described here regard measuring *how much* of a resource is being used, or workload is being applied, and how long to apply the workload to obtain statistically valid results *(initiation and termination)*. Categories A10, A11 and A13 regard instrumentation *(observation)* of those variables considered (by the researchers concerned) to be reliable predictors of power consumption.

Resource instrumentation in microarchitecture and system software (A10) [48,49,56,58,59,61–64,69,73,75–79,81–87] includes approaches that measure resource use. These measurements are then used to predict power consumption. We make a somewhat weak distinction, for reasons we shall refer to, between instrumentation of microarchitecture and instrumentation by system software. The former regards parts of the processor interface that address the processor's infrastructure for monitoring, event counters (e.g., instructions retired, last-level-cache (LLC) misses and translation lookaside buffer misses) and, more recently, power counters (e.g., Intel's Running Average Power Limit (RAPL)) [98]. On the other hand, system software's instrumentation is carried out through intermediary system software and includes, most notably, processor utilization. We have seen references to these two categories as "hw counters" and "sw counters", respectively [99]. The distinction is weak since system software is increasingly exposing microarchitecture instrumentation (consider, for example, Linux's *perf* tool). This reduces the need to directly access hardware registers and blurs the separation between what is abstracted and what is concrete, raw, hardware data. Nonetheless, through our understanding of the data used, we have been able to separate the approaches into two sub-categories.

Category A11 regards the use of a simple characterization of workload as a predictor of power consumption [48,49,53,64,65,67,68,71,78,83,85,87]. This is notably different from approaches in category A10, which are concerned with resource use as a predictor of

power consumption. Other examples (apart from those given earlier) of workload metrics as inputs are the "number of processes" [71] (from the same process image), "transmission rate" [68] and millions of instructions per second (MIPS) [64].

Characterizations may need to be sharper. For example, since packet network traffic arrival is known to often have the properties of a Batch Markov Arrival (stochastic) Process (BMAP), this is an operating constraint (*selection, configuration* and *interconnection* are all ingredient activities here) adopted in the approach of several works studying power efficiency of network functions [83,87,94,100].

Category A12 refers to the measurement of (host) power consumption [48–50,53,55–59,61–64,67,68,71,73,76,79,84] *(observation)* which is most usually measured at the wall outlet or at the power supply inlet. More granular approaches are desirable, and indeed we do find cases [62] that attempt measurement at the power supply output. The principal drawback of such granular techniques is not (principally, at least) construction of intermediary hardware (e.g., riser boards or line resistors) but the difficulty in attributing power drawn through any single dc voltage output (or group thereof) to particular consumers. With the advent of RAPL and certain guarantees on its accuracy, the need for direct power measurement has been, at least partially, avoided.

Category A13 regards the use of hardware sensors to obtain inputs and/or parameters for the power model [60,81,82,87]. These include:

1. Voltage sensors (processor supply voltage);
2. Temperature sensors (processor package and memory temperatures);
3. Fan speed sensors (processor and chassis fans);
4. Wall-clock time measurement.

Some of these variables are used in models that predict power consumption while accounting for the effect of the drift of temperature and automated supply voltage adjustment (in dynamic voltage and frequency scaling—DVFS).

2.2.3. Model Validation

Model validation is a multi-faceted endeavor and this is reflected in the approaches we have detected. The approaches range across the candidates that would typically be considered: simulation (A14) [87], use of test data (A15) [57,62–64,77,85] and corroboration through experimentation (A16) [83,87]. We skip elaborating on these categories as they are either self-evident (A15) or because they are too rarely used to permit general commentary. However, to these three categories we add a fourth (A17), namely, the model adaptation technique, which we describe below and explain why it fits within this branch of the taxonomy.

Model adaptation technique (A17) [57–59,61,62,67–69,71,73,75,76,78,79,81,82,84,86]: This refers to the approach(es) taken (if any) to develop an adaptable model or modeling system. Here, *adaptability* refers to the fitness for use which the model exhibits under changes in one or more of the seven dimensions of variability that will be defined in detail in Section 3.1.1 below. Model adaptability is essential for practical virtualization, where changes in, for example, the number of co-hosted, concurrent VEs, or in workload type, are commonplace. Here, we list the major approaches taken towards producing adaptable models. Since these approaches emerge in the context of validating a model's accuracy under some limited range of the seven-dimensional space of operating conditions, we classify this category of approaches as an aspect of model validation.

1. *Adaptation to change in the number of co-hosted, concurrent VEs* is widely achieved through the *time-division multiplexing* of event counters [59,62,69,79,86], RAPL counters and CPU utilization [61,73]. This approach enables the use of such metrics as predictors of dynamic (active) power consumption, by apportioning counts to VEs in accordance with the time during which the VEs were active.
2. *Adaptation to uncorrelated causes of power consumption* can be achieved through *additional predictors* [82] to follow causes of power consumption that do not correlate well with counters within the current set. This case reflects itself as poor accuracy in

predicted power consumption. Although counter-based models are reported to fit a variety of processor- and memory-intensive workloads well, it may be necessary to account for unanticipated activity through the approach of adding previously unused counters.

3. ***It may not be possible to fit a single model with parameters known* a priori**, to the whole range of inputs within the scope of study, notwithstanding the diversity of predictors employed in this pursuit. The following adaptive techniques have been found in the literature.

   a. *Dataset partitioning*, where the dataset is of the form { {predictors}, response } and is used in [77] to match the best model out of a set of models to the current, actual operation. An individual model in the set is associated with a single node in a decision tree and the node is selected according to features pertinent to the current, actual operation. A simpler, but conceptually similar, approach is taken in [58]. A number of models are devised and model-selection features are limited to the number of active VMs and a coarse grading of CPU utilization.

   b. *Modeling on demand* is the term we use to succinctly refer to the fourth adaptation class of techniques:

      i. One early example of this approach is found in [75], where the dependence of the model on the workload has been addressed through ***online training***, whenever prediction accuracy of the extant models falls out of a range of tolerance. The rationale adopted is that if model adaptation to such an "unseen" case is limited to parametric tuning, then a modeling system might be able to construct a model while VEs are in operation;

      ii. A broader perspective is found in [78]. An automated system for profiling containerized applications is described and demonstrated. Containerized applications are profiled from three perspectives: computing resources consumed, energy consumed and performance. In this case, the rationale is that energy consumption can be optimized by the determination of a frequency-and-hardware-threads host configuration that meets performance requirements. Thus, starting from central functional requirements (performance requirements), operating conditions are determined that minimize energy consumption. This approach is capable of meeting the challenges posed by heterogeneous host hardware and application (workload) diversity, at the cost of analytical modeling. Indeed, characteristic curves can be derived but causes underlying observed behaviors remain unaddressed.

2.3. A taxonomy of Developments

Developments fall cleanly into one of two groups: (a) models of power consumption and (b) observations on dependencies of power consumption. The first group (D1–D10) includes developments that ***predict power consumption*** over a sub-space of the seven-dimensional space of operating conditions. The second group (D11–D18) includes developments that are oriented towards the ***correlation of power consumption*** with aspects of system integration. As the taxonomy is rather broad, we present it in three parts:

1. Figure 3: top-level fork into models and dependencies;
2. Figure 4: the taxonomy of models;
3. Figure 5: the taxonomy of dependencies

D1–D4: We first present four categories of developments that concern ***models of host-system power consumption*** characterized by the condition where ***workload is processed by VEs***:

- D1: linear regression models;
- D2: non-linear regression models;
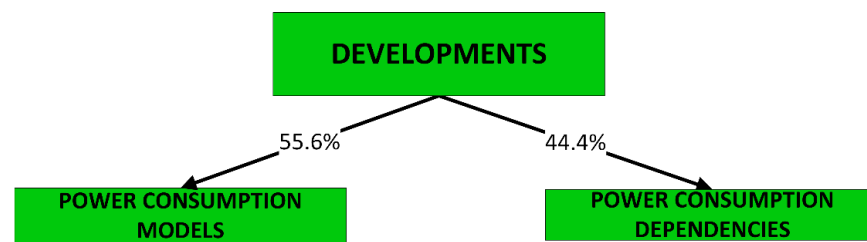- D3: machine-learnt models;
- D4: models of local mass storage.

Figure 3. A top-level division of the developments, with frequency of occurrence shown as a percentage.
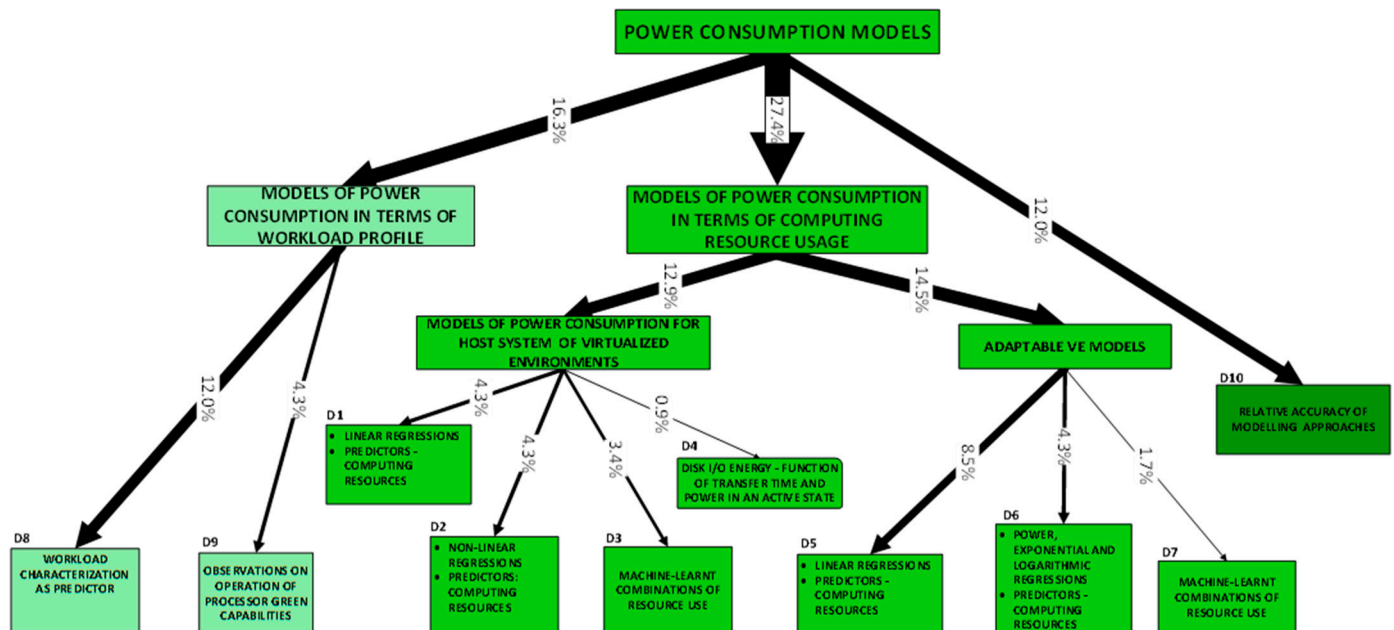

Figure 4. A taxonomy of power models, with frequency of occurrence shown in line thickness and as percentage.

This contrasts with the scope of developments referred to under categories D5, D6 and D7, where *models of VE power consumption* are presented.

Models in these categories are interesting from the perspective of analyses of *sets of hosted VEs* that seek to identify operating conditions of optimal host power efficiency. As predictors, such analyses use instrumentation that measures resources used by the VEs. Categories D1, D2 and D3 all predict power consumption in terms of resource use but differ in the type of model produced.

- D1 regards models of power consumption through linear combinations of scalar predictors [57–59,69,75]. The scalar predictors are resource usage metrics.
- D2 regards polynomial or simple mathematical powers of resource use (the scalar predictors) [57,60,61,76,81].
- D3 regards models that employ machine learning (e.g., Gaussian Mixture, Support Vector Machine, Neural Networks) [62,77,79].

Category D4 regards the models of power consumed by mass storage local to the host system [86]. These models attempt to predict power consumption in terms of activity metrics such as the total amount of time spent in a known state (in terms of power consumption, e.g., active/idle), the rate of data exchange (MB/s or input/output operations per second) and the mode of operation (sequential/random and read/write). In the context of the approximations observed in the development of these models, their accuracy cannot be fair.
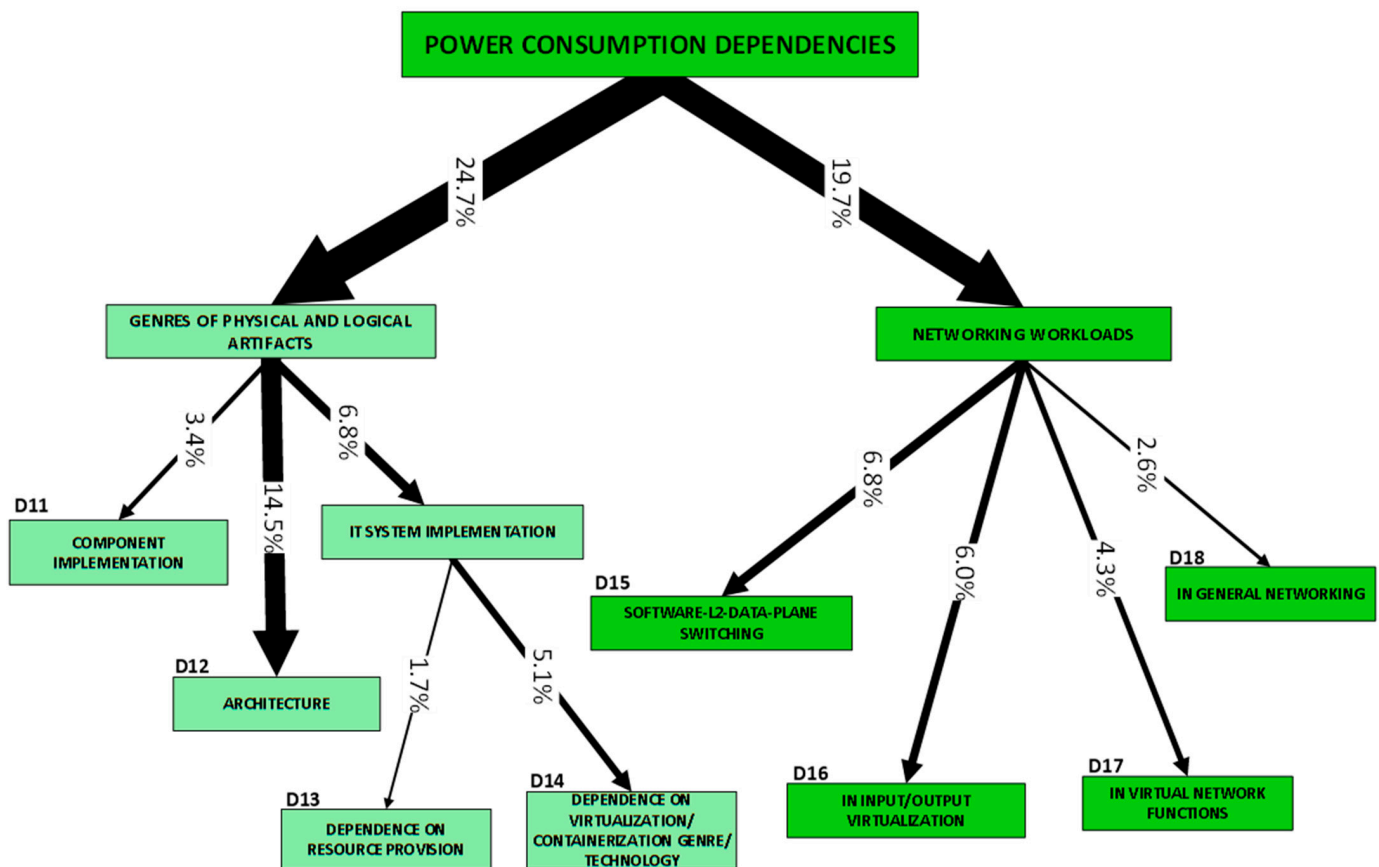
Figure 5. A taxonomy of power dependencies, with frequency of occurrence shown in line thickness and as a percentage.

2.3.1. Models of Power Consumption

Adaptable VE models (D5–D7): Developments within these categories consist of *adaptable* models of the virtualized entity's power consumption and have two important characteristics in common:

1. They are adaptable to a variable number of concurrent, co-hosted (active on the same host system) VEs (we refer to the latter scope of variability as the seventh dimension of variability);
2. The predictors are the measured amount of computing resources used by the VEs.

Models can be distinguished by the predictors they use, workloads employed and modeling approach:

1. *Predictors* (of VE power consumption) are obtained from system software's instrumentation, e.g., CPU utilization (see approach categories A10–A13), and from microarchitecture instrumentation, e.g., LLC misses (again, see approach categories A10–A13);
2. *Workloads* used to obtain the model (this restricts the range of workloads within which the model is valid) may be:

    a. Specific workloads: the most restrictive, as they relate to a particular test load;
    b. Synthetic workloads: less restrictive than specific but limited to the exercise of one resource, typically the CPU;
    c. Combinational workloads: still less restrictive, involving the exercise of a number of resources of the host system (e.g., SPEC CPU benchmarks may be both processor and memory intensive);
    d. Representative workloads (e.g., TPC-W [101]) produce models that are readily associated with use cases.

3. The modeling approach may be:
    a. Linear regression (category D5) [58,59,69,73,75,81,82,84,86];
    b. Power (integer- and non-integer powers), exponential and logarithmic regressions (category D6) [61,66,76];
    c. Machine-learnt combinations of resource use (category D7) [62,79].

Models of power consumption that use workload profile as predictors (D8, D9): Categories D8 and D9 group developments from (two) sets of RUs that predict power consumption of hosts and/or VEs through (the measurement of) some characteristic of the submitted workload. This contrasts with RUs in categories D1–D7, where prediction is obtained through (the measurement of) some computing resource (processor and/or memory and/or I/O). Most developments in this category are obtained through the abstraction of hardware by one or more model parameters that express power consumption under case-specific conditions of operation. Some of these abstractions are identified in the descriptions of these two categories.

Developments in D8 [63–68,71,78,83,85,87,94] use:

1. Processing load (number of processes, millions of instructions per second (MIPS), etc.) pertaining to a specific application, as predictors of *host system* power [63–67,71];
2. Packets per second, through an intrusion detection system implemented in a VNF [85];
3. Transcoded frames per second, through a transcoder implemented in a containerized network function (CNF), and inferred images per second, also in a CNF [78];
4. Average network transmission rate, as a predictor of host system power [68];
5. Statistics of a Batch Markov Arrival Process (BMAP) (packet traffic) as a means of the prediction of power consumption by a VNF [83,94].

Hardware is abstracted through the measurement of power consumption at some operating point (a specific operation is being carried out), or a change in power consumption over some operating range. Examples follow:

1. In [65], where energy efficiency of an interactive web service is studied, the operating point is an entire VM running the TPC-W benchmark [101].
2. In [67,71], the operating points are the host's power consumption when (a) idle, (b) one core is active (processing load) and (c) maximum, with all cores active. Furthermore, use is made of the step increment in consumption corresponding to the activation of each additional core. Cores are activated when they are utilized by VEs.
3. In [68], the operating point is the power consumption when co-hosted VEs are transferring a file to a client computer. An *affine* relationship between the host's power consumption and its transmission rate (transmissions originate on hosted VEs) is found.
4. In [64], an operating range is used: the increase in power consumption that corresponds to an increase in MIPS on the VEs.
5. In [83], the operating point is the power consumption when a VE running on a single processor core is switching packets at the maximum rate for a given performance state.
6. In [85], fifty-four (54) different features of network traffic are input to an artificial neural network that selects the operating frequency that optimizes power consumption.

With one exception, none of the works in the above list uncovers the hood to peer at the processor's internals (to obtain predictors of power consumption). The exception is [83]; yet even in this case, the performance monitoring counters are not used as direct predictors of power consumption, but to obtain (a) the timing information necessary for a queueing model and (b) the operating state (ACPI (Advanced Configuration and Power Interface [102]) P- and C-states).

Developments in D9 [87] are set within the models branch of the taxonomy. These developments may be considered as useful observations on *the operation of processors' green capabilities*. Examples of these observations (all from [87]) include:

1. Low-power instructions might be a better candidate than low-power idle to save power under higher link utilization;

2. Operation in full ACPI P-state, operation with low-power instructions on idle detection and operation with low-power idle on idle detection are (a) in ascending order of latency to return to active processing and (b) in descending order of power consumption;

3. A processor utilization threshold below which low packet latency is guaranteed under BMAP traffic arrival is 80%.

By "green capabilities", we refer to a broader range of microarchitectural aspects than the by-now-conventional adaptive rate and low-power idle operation. While these latter two remain at the center of attention, there is also the means of low-power instructions [87] that has been successfully employed to improve power efficiency. Notwithstanding the origin of these observations in modeling work, it may be argued that they might also be classified within the dependencies branch of the taxonomy. We have chosen the models branch, but as further studies add to the body of knowledge on how to operate processor green capabilities, this category's position in the taxonomy might need to be changed.

Relative accuracy of modeling approaches (D10) [60,62,63,77]: Developments presented under category D10 are comparisons of the relative accuracy of alternative modeling methods with respect to conventional polynomial (including linear), power, exponential and logarithmic regression. These developments have been found within works that show models classified under category D9. The purpose is to qualify and quantify improvements of machine-learnt models with respect to conventional regression models.

2.3.2. Dependencies of Power Consumption

This parent node of the taxonomy is divided into two child nodes that are not strictly mutually exclusive. For example, the software data plane is considered in works under D15. Clearly, the software data plane is a logical artifact and might be included within a child node of "physical and logical artifacts", or directly thereunder as a leaf node (i.e., as a category). The choice of separation of D15–D18 and the inclusion under the parent node "networking workloads" was taken for two principal reasons. Firstly, the recurrence of investigation of power consumption's dependency on networking workload merits attention through separation. Secondly, as this survey caters for an audience with an interest in softwarized networking, an emphasis on the power dependency on networking workload seems justified.

Knowledge about the dependency of power consumption on specific hardware (D11); the dependency of power consumption on architecture (D12); the dependency of power consumption on resource provisioning (D13); and the dependency of power consumption on virtualization genre and technology (D14): Categories D11–D14 are grouped into a set of works that obtain the sense of the correlation (positive/negative/none) between power consumption and some genre of artifacts:

1. Specific hardware types (D11);
2. Computer architecture (D12);
3. Resources provided (D13);
4. Virtualization genre and technology (D14).

Category D12 groups works that relate to observations on the impact of architectural features on power consumption [49,50,67,71,76]. While these observations are useful, they are generally too coarse to be directly applicable to real-time power control. Their use emerges from guidance which they provide in the development of power models. For example, it was observed that when the number of threads that fully occupy a core's time (active 100%) exceeds the number of logical cores in the system, the energy efficiency (measured, in this case, in hash/J, a computational metric of energy efficiency) decreases [50]. Evidently, this is good guidance; equally evidently, it is not a directly applicable development.

Categories D11, D13 and D14 relate closely to (various aspects of) implementation, and as such are of particular interest to the *system integrator*. Data of high quality from these developments inform and guide the tasks of gathering components into sys-

tems that meet the non-functional requirements obtained from concern with energy and power efficiency.

Category D11 gathers observations about the dependency of power consumption on specific processor hardware [49,69,76]. We have observed that these developments are gathered as a by-product of the process of research; they are rather incidental. Like any implementation, their usefulness is limited to the lifetime of the concerned device(s).

Category D13 [73] includes developments that regard the specific resource configuration of:

1. The instantiating host, i.e., the relationship between a VE's power consumption and the specific resources of its host hardware specifics such as the number of cores and amount of memory carried by the host instance;
2. The guest VE, i.e., the variation of a VE's power consumption with its resource assignment, on the same host.

Category D14 gathers observations about the dependency of power consumption on instances of virtualization genre and technology [55,56]. Developments in this category are less incidental than those in category D2 and are obtained with the focused intention of tackling challenges relating the power consumption of implementations. These developments relate to a less diversified group of implementations. For example, there are fewer virtualization platforms than processors to choose from. Direct use of these developments is mostly limited to the specific implementations concerned; however, some generalizable conclusions exist. For instance, it was observed that both hardware-assisted virtualization and paravirtualization are less efficient (in the specific empirical setup) than non-virtualized operation in the use of processor caches [56]. This empirical evidence favors the hypothesis that cache hit ratios suffer due to the greater thread rotation in virtualized and containerized environments.

Knowledge about dependency on power consumption while processing a networking workload (D15–D18): Development categories D15, D16, D17 and D18 are grouped here as their central characterization is knowledge about the behavior of power consumption by VEs while processing a networking workload. Categories D15, D16 and D17 reflect the challenges described in P11, P9 and P10, respectively.

Category D15 regards contributions to knowledge about the power consumption of software data planes [48,53]; D16 regards virtualization of network I/O [51,53], and D17 is about network functions [48,53,85]. Thus, in [51], the power efficiency of DPDK PMDs is demonstrated with respect to Netmap drivers, for packet transmission. This development is balanced by [48], where the power efficiency of transmission through a DPDK-enhanced Open vSwitch is shown to be worse (@500-byte Maximum Transmission Unit (MTU)) than that of the unenhanced Open vSwitch. In each of these categories, efforts are made to allocate burden through the isolation of power consumption and the attribution to the sub-system (data plane/virtualized IO/network function) under study.

Category D18's developments differ from those of D15–D17 because they cut across these categories' sub-system boundaries [56,65]. For example, in [65], the energy efficiency of web transactions executed on co-located VMs is found to be highest in the operating condition of processor-core over-subscription (more VMs than cores). In [56], it is shown that power consumed by packet delivery to a VM through a software packet switch is much higher than that required for delivery in a non-virtualized environment. In both these papers, the object of interest incorporates the virtualization of network I/O *and* the data plane. In [65], the object of interest encompasses the network function: a web service and accompanying application and database components.

All four categories are of keen interest to the system integrator. The emphasis is on the components in the integrator's set of building blocks, specifically on the behavior of power consumption of various implementations, and types thereof (in the scope of the categories).

2.4. P–A Dyads (Problem/Challenge-Approaches) Graphics

Over the following pages, we present a comprehensive set of graphics (Figures 6–17) that illustrate the approaches detected to tackle specific challenges (reference numbers for pertinent RUs appear in the shape tags). For example, consider Figure 7. The graphic shows the component approaches applied to discover "the impact of specific architectural attributes of the host system on power consumption of VEs" (see problem category P1).

Figure 6. Approaches to solving challenges in category P2; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P2: [55,56]; A6: [55,56]; A8: [55,56]; A9: [55,56]; A12: [55,56].

The presence of approach categories A1, A5, A6, A8 and A10–A12 does not mean that *every* RU tackling this challenge uses a component from *all* of the approach categories. It does, however, mean that every RU tackling P1 uses a subset of the approaches shown. We proceed by giving examples (relevant to P1), with references.

1. *Managed resource provision (A6):* Prior to running experiments, researchers set the conditions for the experiment through this approach [48–51].
2. *Resource-specific workload (A8):* This may be used to stress the component implementing the architectural attribute under test [48–51].
3. *Simple workload characterization and instrumentation (A11):* The workload must be characterized by some parameter that serves to measure its demand for power [48,49].
4. *Resource instrumentation (A10):* This is an alternative to the use of workload profiling (A11) as a predictor of power consumption. Rather than use, say, the number of threads, or transmit bandwidth (for networking workloads) as predictors, this approach uses resource instrumentation [48,49].
5. *Identification of metric of energy efficiency (A5):* In certain cases [49,50], energy or power efficiency is investigated, rather than energy or power consumption. In these cases, the researchers identify and use a relevant metric of efficiency, rather than metrics of consumption (watts or joules).
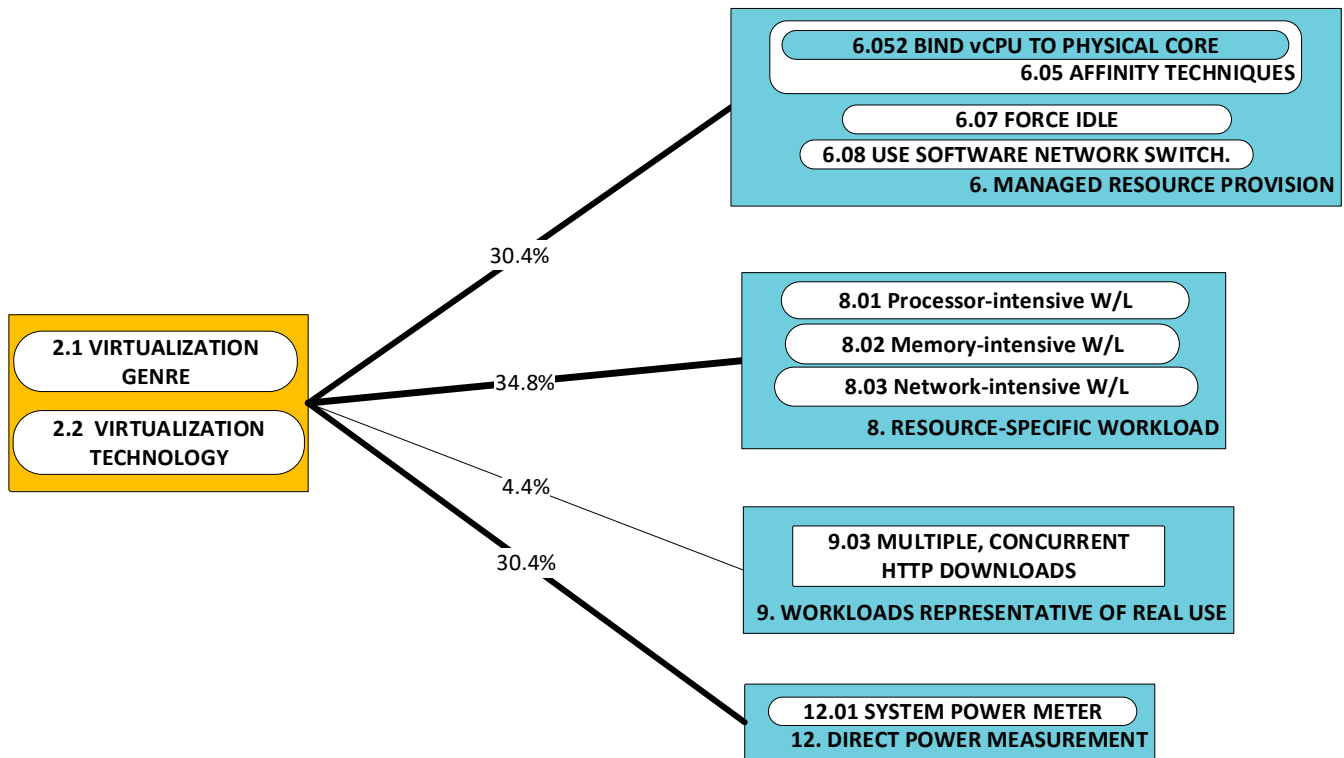
Figure 7. Approaches to solving challenges in category P1; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P1: [48–51]; A1: [48]; A5: [49,50]; A6: [48–51]; A8: [48–51]; A10: [48,49]; A11: [48,49]; A12: [48–50].

Figure 8. Approaches to solving challenges in category P6; utility metric U_{A_k} shown in line thickness and as a percentage. RUs in the respective categories are the following: P6: [60,81,82]; A13: [60,81,82].

On any of the dyad graphics, the approaches shown (inside the approach categories) include only those which are used by at least one RU that tackles the challenge category that is the root of the dyads. We delegate a repository with a full tabulation of all approaches in any approach category. Similarly, all individual developments within a category are delegated to the same repository (https://github.com/humaira-salam/PowerMeasurementAndModelingRawData, accessed: 30 October 2022).

2.5. Causality DAG

The Causality Directed Acyclic Graph (DAG) in Figure 18 shows a bird's eye-view of the proceedings of research in scope.

Figure 9. Approaches to solving challenges in category P3; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P3 [57–69,71]; A2 [57–64]; A5 [65]; A6 [60,66–68]; A8 [59–61,63,64,66–68,71]; A9 [57,58,60,62,65,68]; A10 [57–59,61–64,69]; A11 [64,65,67,68,71]; A12 [57–59,61–64,67,68,71]; A15 [57,62–64].

Figure 10. Approaches to solving challenges in category P10; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P10: [48,53]; A6 [53]; A8 [53]; A9 [48]; A11 [53]; A12 [48,53].
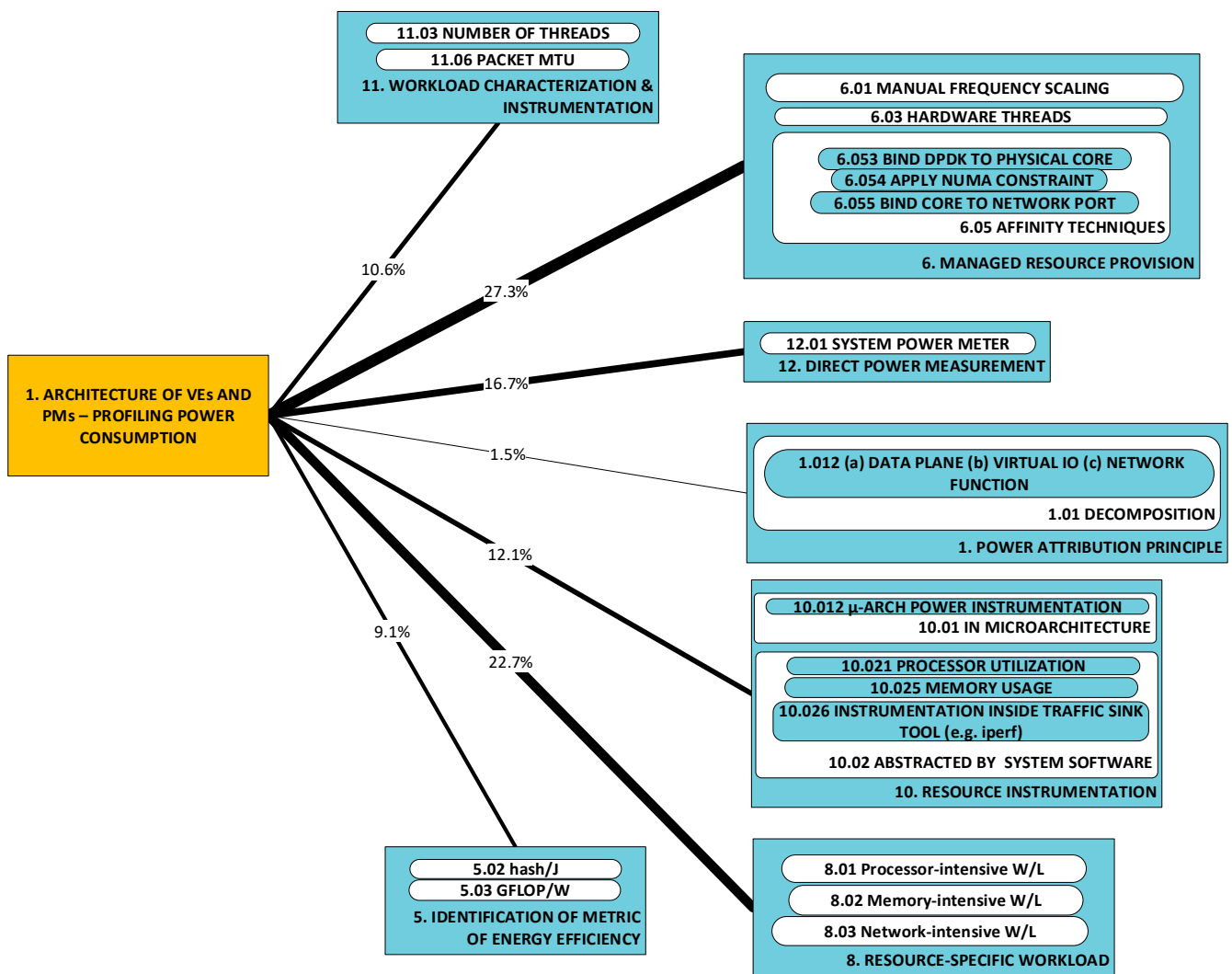
Figure 11. Approaches to solving challenges in category P4; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P4 [56,62,67–69,71,75–79]; A1 [62,77]; A2 [62,77]; A7 [78]; A8 [56,67,69,71,75,76]; A9 [56,62,68,75,79]; A10 [62,78,79]; A11 [78]; A12 [56,62,76,79]; A15 [62,77]; A17 [62,67,68,71,75,76,78].

Figure 12. Approaches to solving challenges in category P11; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P11 [48,53]; A1 [48]; A8 [48,53]; A10 [48]; A11 [48,53]; A12 [48,53].
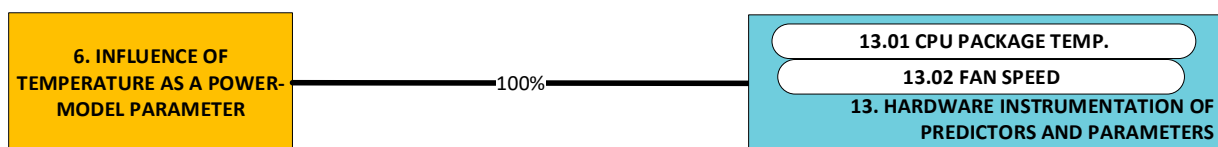
Figure 13. Approaches to solving challenges in category P5; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P5 [62,72,78]; A1 [62]; A2 [62]; A6 [72]; A7 [78]; A9 [62]; A10 [62,78]; A11 [78]; A12 [62,72]; A15 [62]; A17 [62,78].
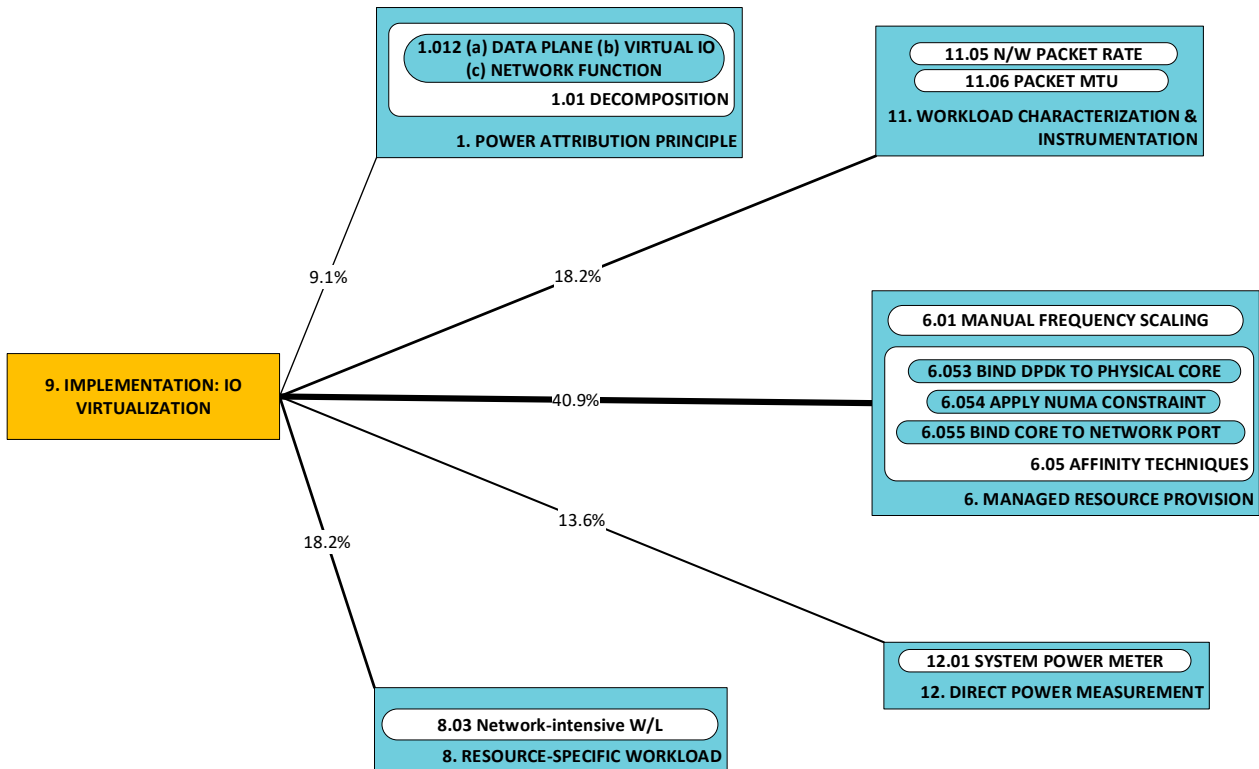
Figure 14. Approaches to solving challenges in category P9; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P9 [51,53]; A1 [53]; A6 [51]; A8 [51,53]; A11 [53]; A12 [53].
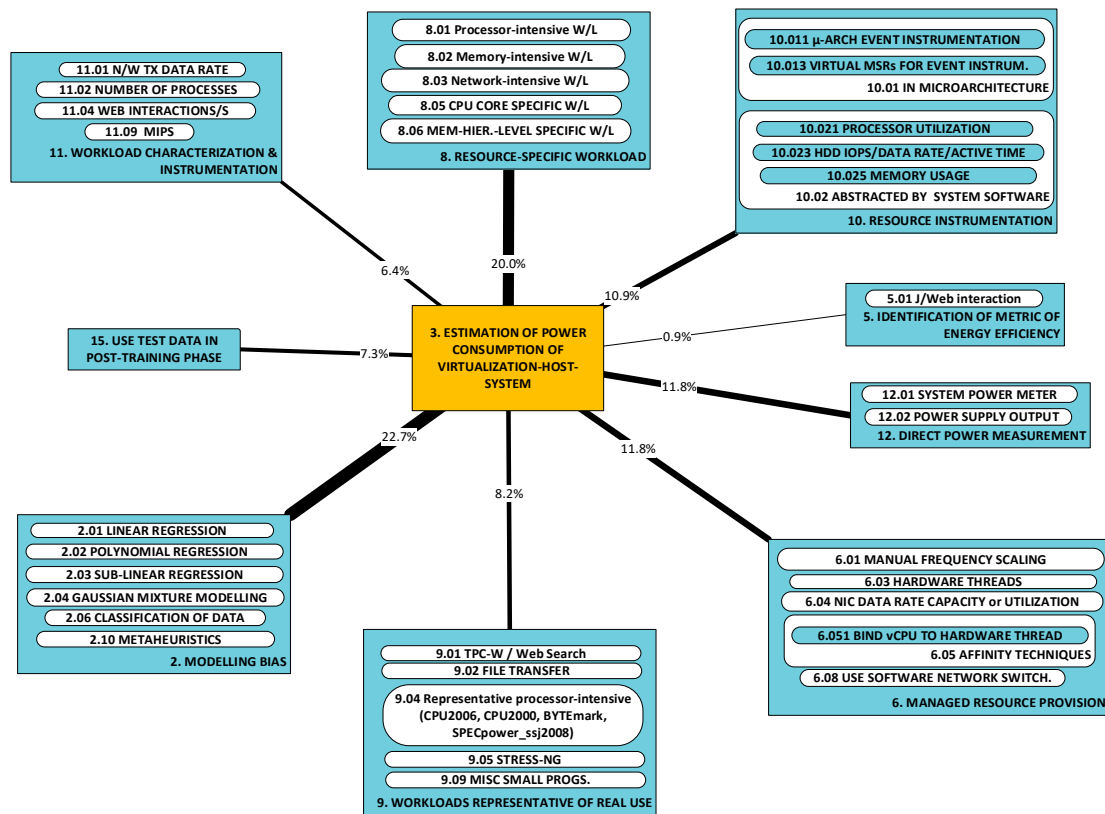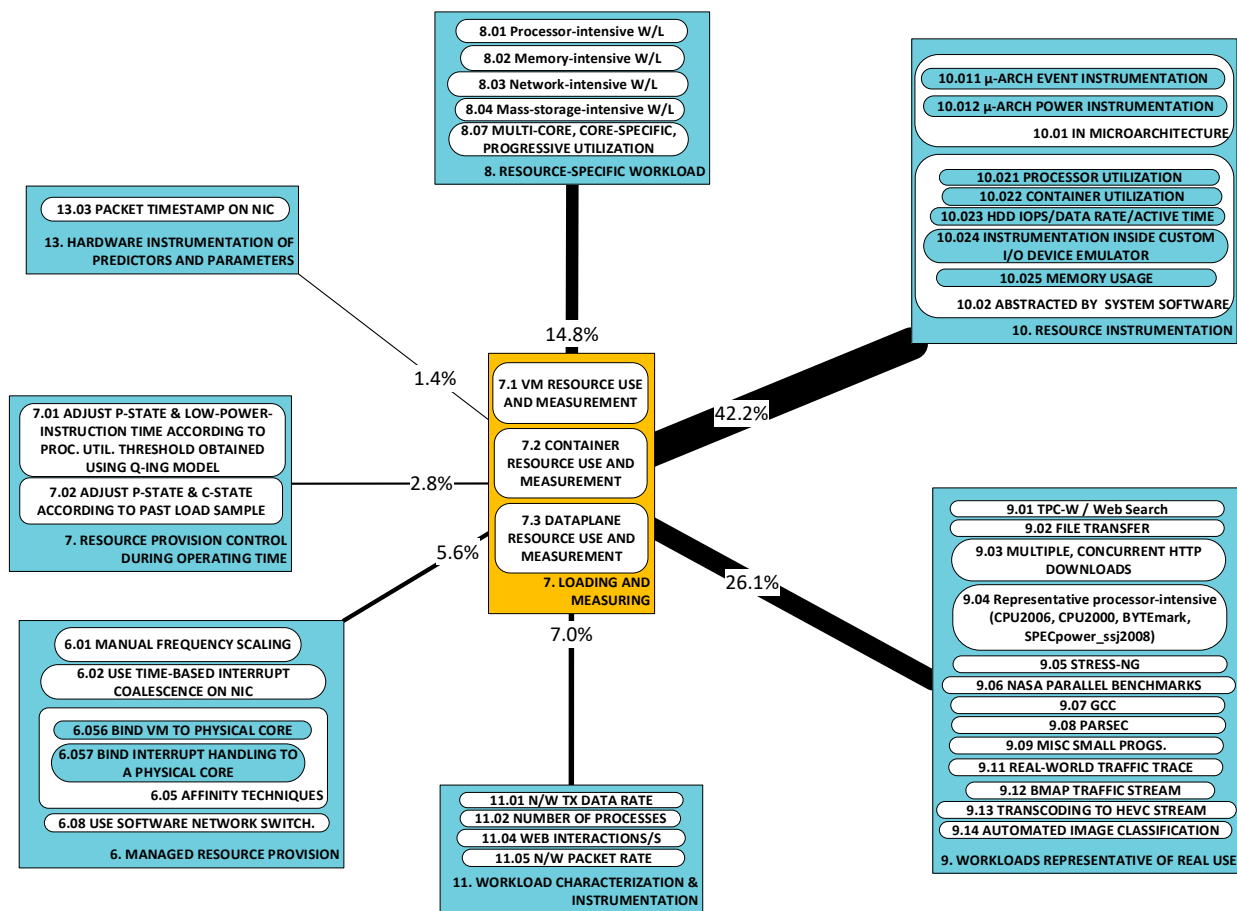
Figure 15. Approaches to solving challenges in category P7; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P7 [56–59,61,62,65, 68,69,71,72,75–79,81–85,87]; A6 [56,68,81–83]; A8 [56,59,61,71,76,77,84,85]; A9 [56–58,62,65,68,69,72, 78,79,81–84,87]; A10 [56–59,61,62,69,72,75–79,81–85,87]; A11 [65,68,71,83,84,87]; A13 [87].

2.6. Triads (Problem/Challenge-Approaches-Developments) Graphic

Figure 19 shows the triads graphic. To improve readability, we limit illustration to the triads that are in the top twenty percentile of a list ordered according to thickness. These triads comprise 49.2% of the total number of compiled triads.

2.7. Statistics

Bar charts that illustrate the category metrics described in [1, N. see section on "Statistics"] are presented below:

1. Challenges (Figure 20)
   a. Frequency of occurrence in the set of all RUs;
   b. Frequency of occurrence in the set of all challenges in all RUs;
   b. Frequency of occurrence, weighted by approach diversity, in the set of all challenges in all RUs.
2. Approaches: (Figure 21)
   a. Frequency of occurrence in the set of all approaches in all RUs;
   b. Frequency of occurrence in the set of all triads in all RUs.
3. Developments: frequency of occurrence in the set of all developments in all RUs: (Figure 22)
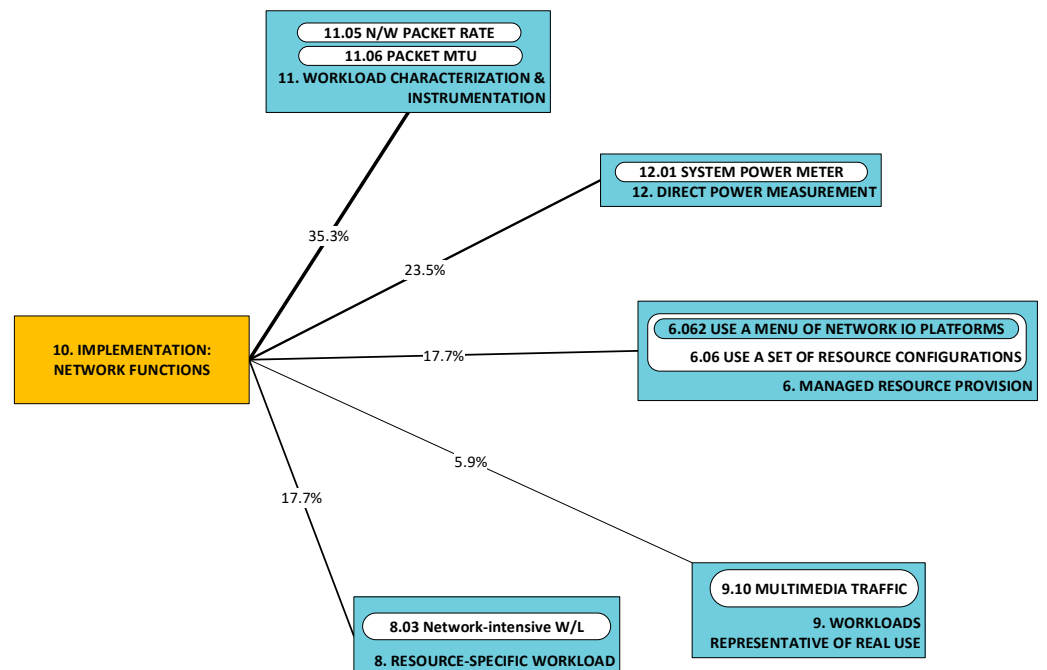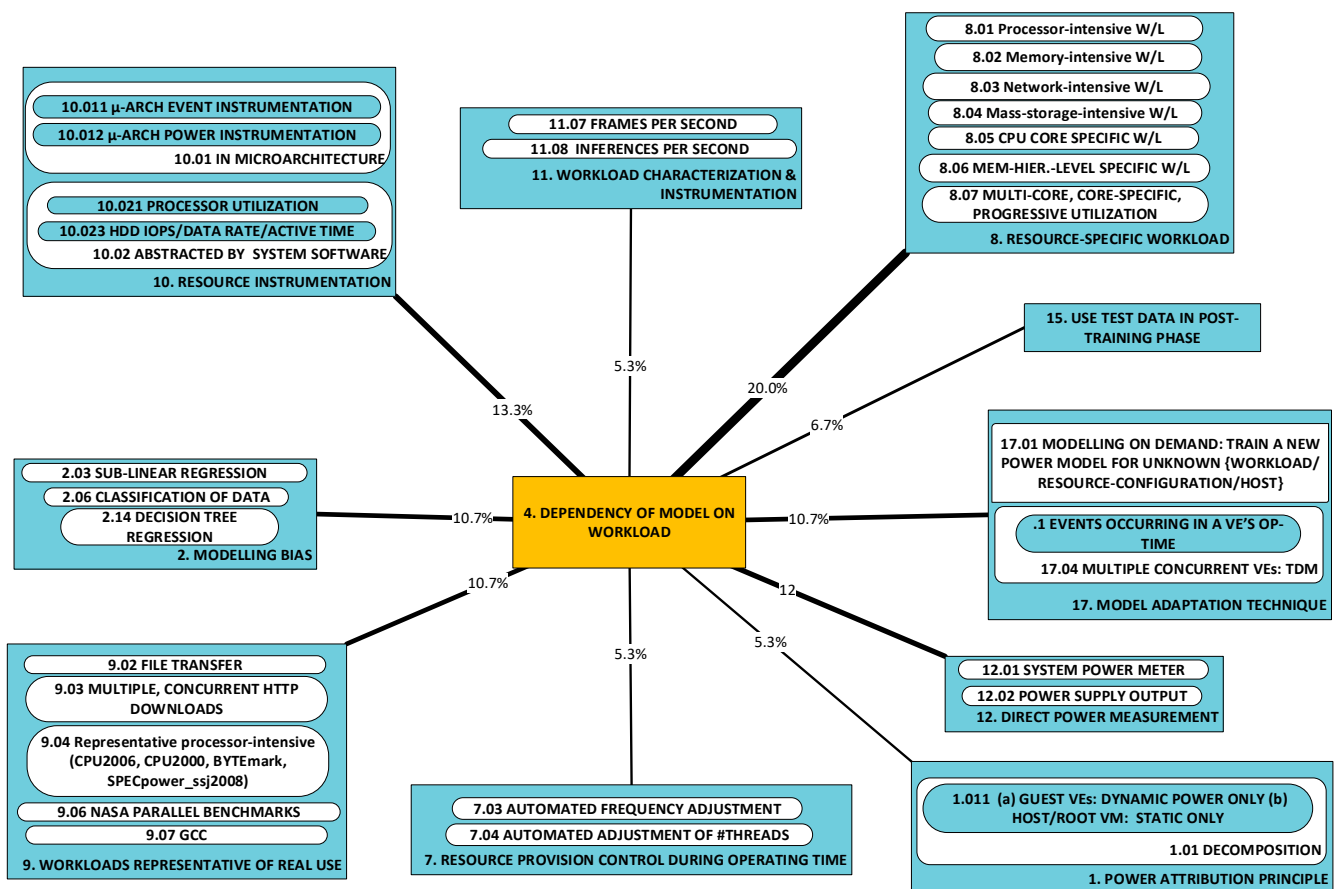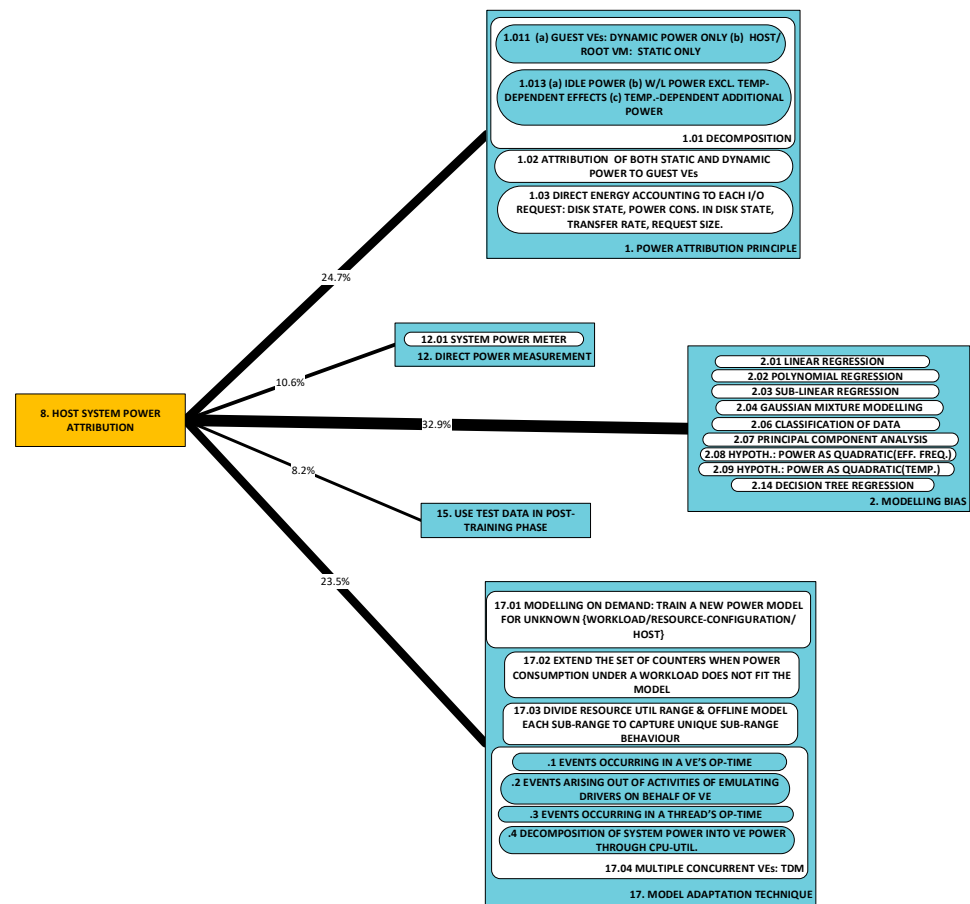
Figure 16. Approaches to solving challenges in category P8; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P8 [57–59,61,62,69,72, 75–77,79,81,82,84,86]; A1 [57–59,61,62,69,72,75–77,79,81,82,84,86]; A2 [57–59,61,62,72,75–77,79,81,84]; A12 [57–59,61,72,76,84]; A15 [57,62,77]; A17 [57–59,61,62,69,72,75,76,79,82,84,86].
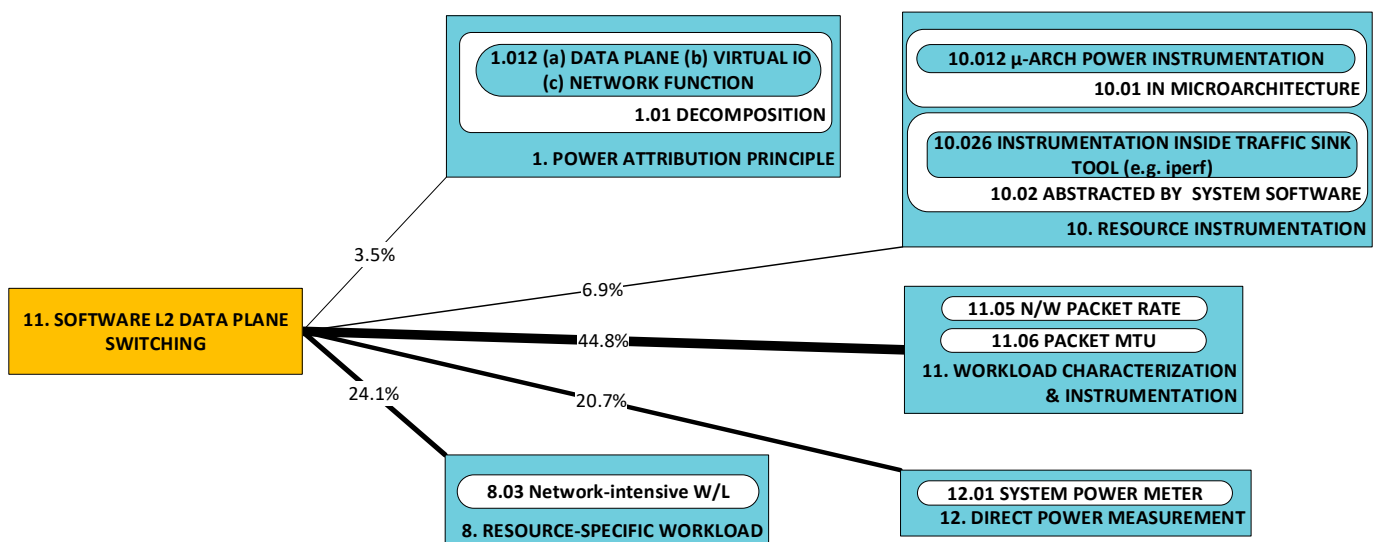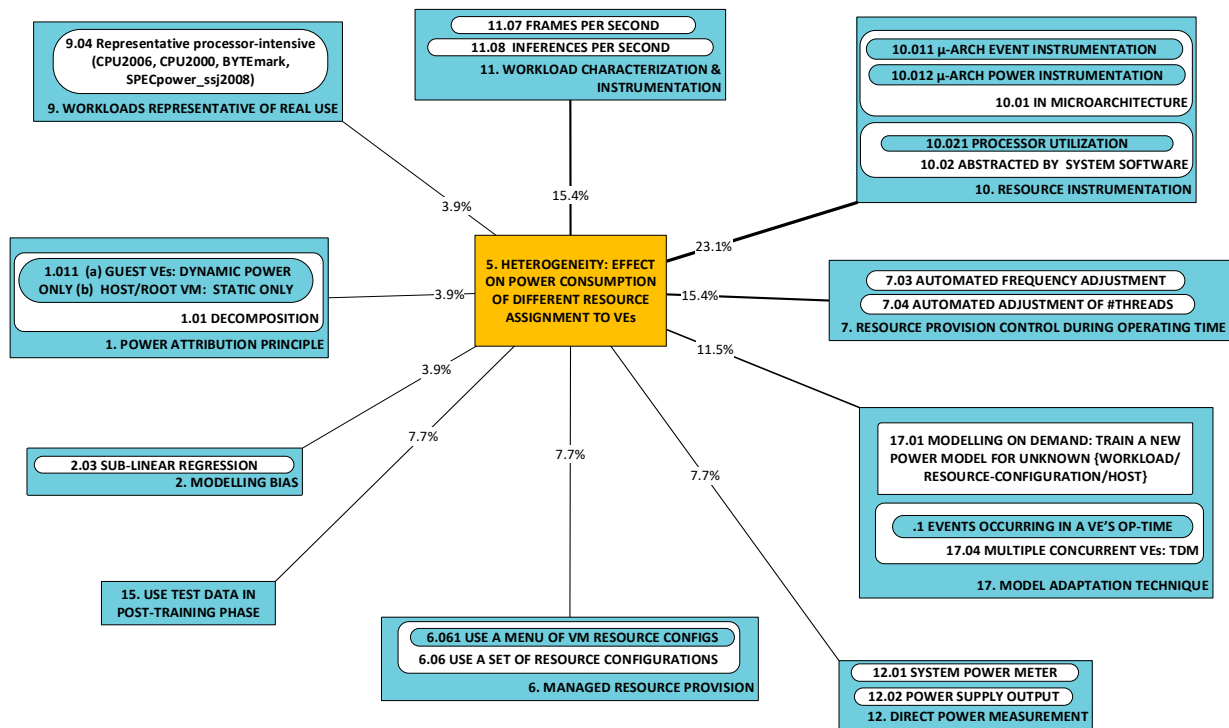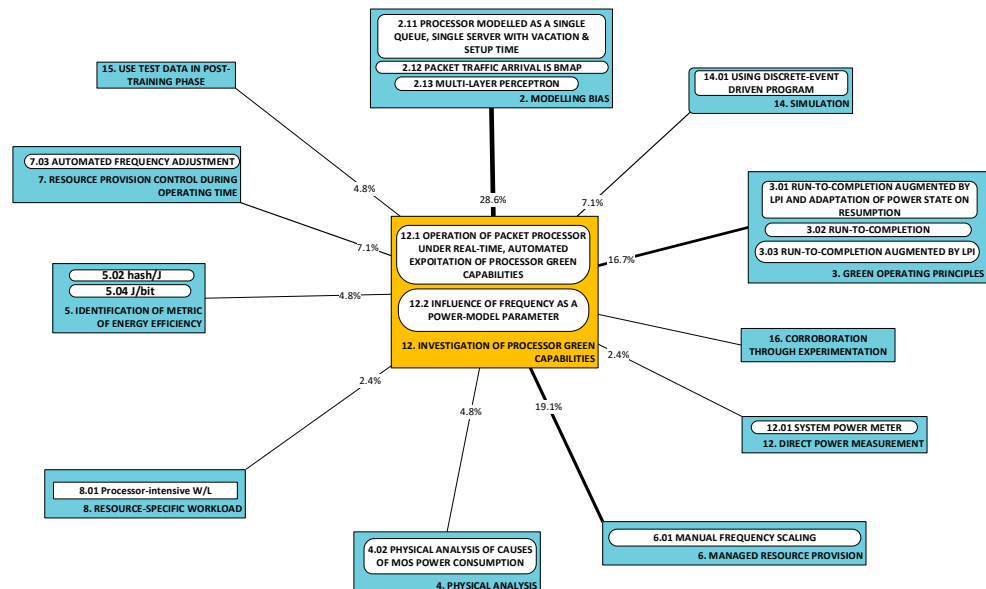

Figure 17. Approaches to solving challenges in category P12; utility metric U_{A_k} is shown in line thickness and as a percentage. RUs in the respective categories are the following: P12: [50,60,67,81–83,85,87,94]; A2 [83,85,87,94]; A3 [83,87,94]; A4 [60]; A5 [50,85]; A6 [50,60,67,81,82]; A7 [85]; A8 [50]; A12 [50]; A14 [87]; A15 [85]; A16 [83].
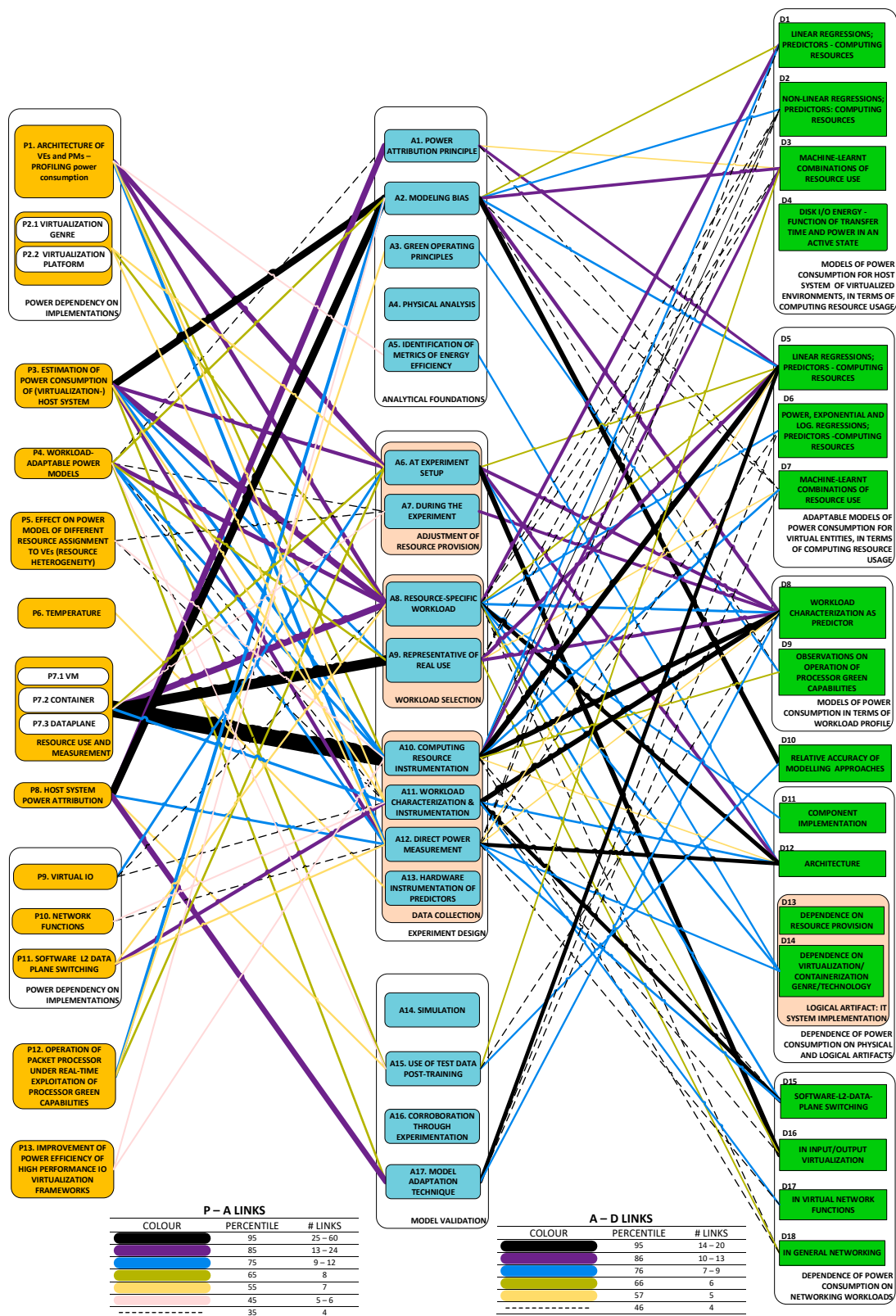
Figure 18. A directed acyclic graph showing the distribution of research into power measurement and power consumption models in virtualized networking and computing environments. D1 [57–59,69,75]; D2 [57,60,61,76,81]; D3 [62,77,79]; D5 [58,59,69,72,75,81,82,84,86]; D6 [61,66,76]; D7 [62,79]; D8 [63–68,71,78,83,85,87,94]; D9 [87]; D10 [60,62,63,77,79]; D11 [49,69,76]; D12 [49,50, 67,68,76]; D13 [73]; D14 [55,56]; D16 [51,53]; D17 [48,53,85]; D18 [56,65].
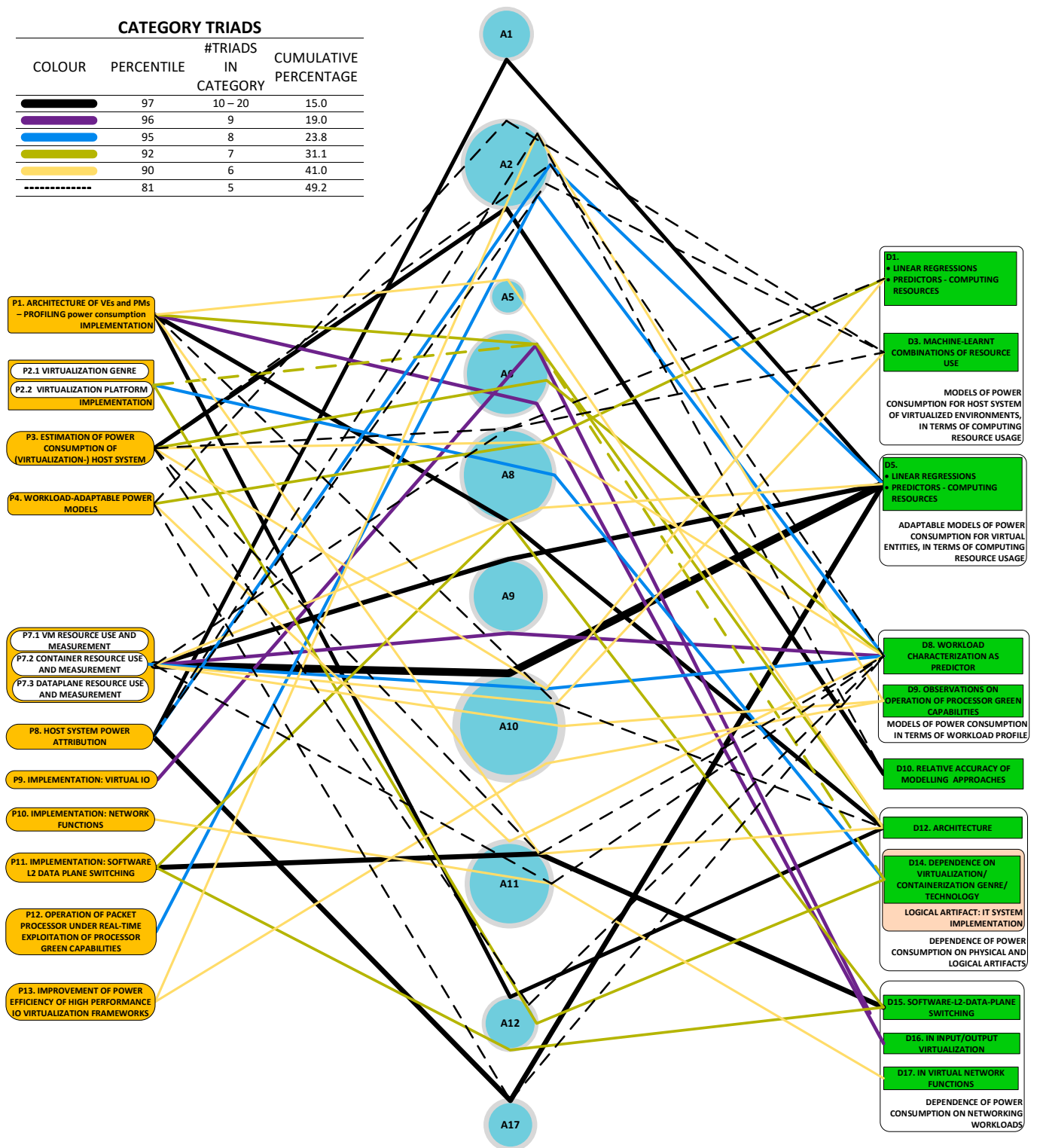
Figure 19. Triads. D1 [57–59,69,75]; D3 [62,77,79]; D5 [58,59,69,72,75,81,82,84,86]; D8 [63–68,71,78,83, 85,87,94]; D9 [87]; D10 [60,62,63,77,79]; D11 [49,69,76]; D12 [49,50,67,68,76]; D14 [55,56]; D15 [48,53]; D16 [51,53]; D17 [48,53,85].
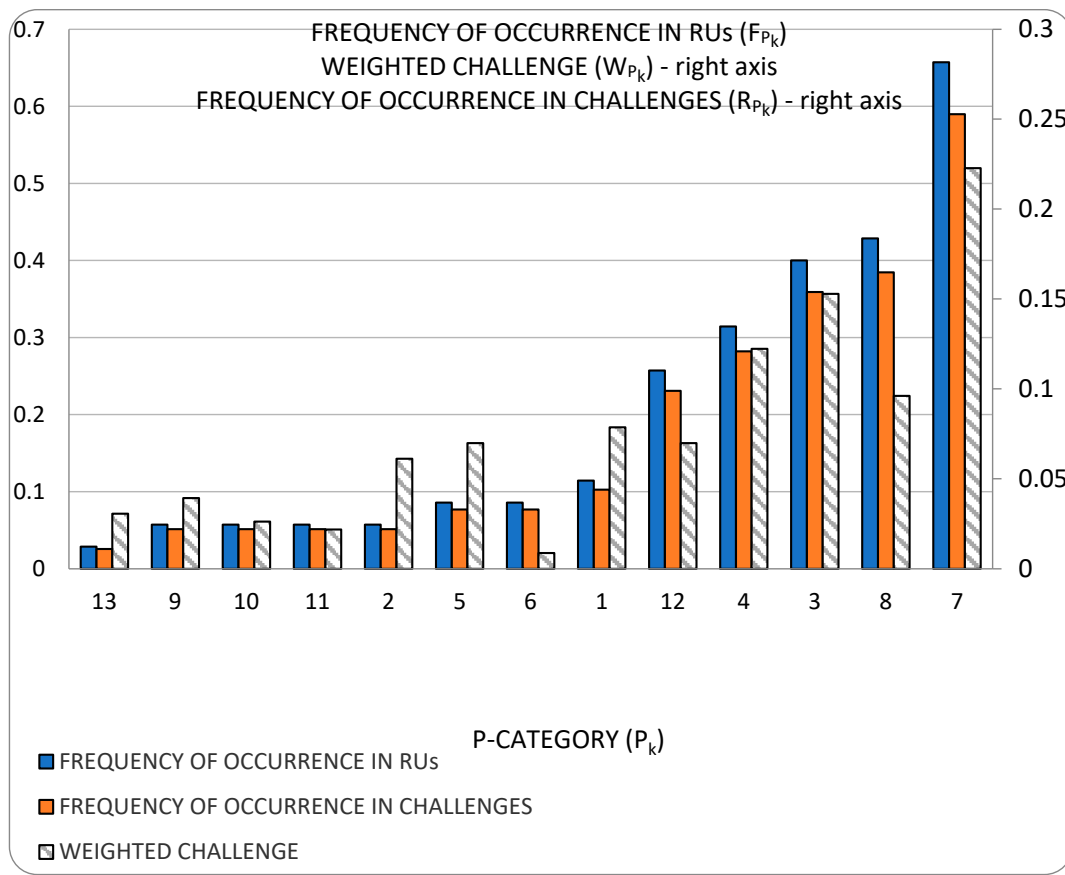
Figure 20. Frequency, Research Interest and weighted Challenge bar chart.
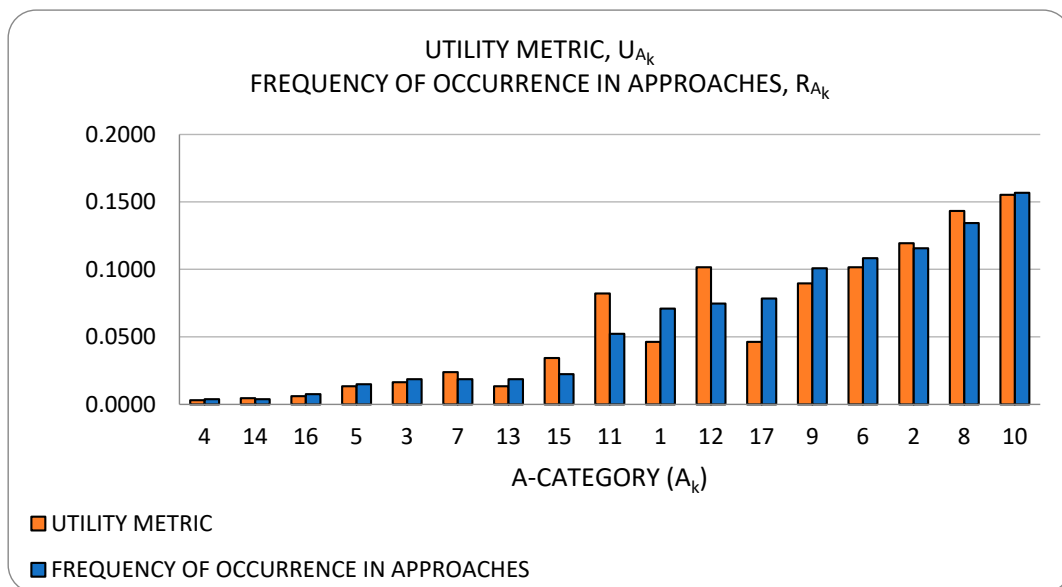
Figure 21. Approach metrics.

To improve legibility, the x-axis labels show category numbers only. The codes ("terse, dense representations of a verbose articulation of a concept", see [1, N. see sub-section "what are codes?"]) linked to the numbers are shown in Tables 2–4. In Table 2, we also show questions that help to clarify articulation of the challenge posed.
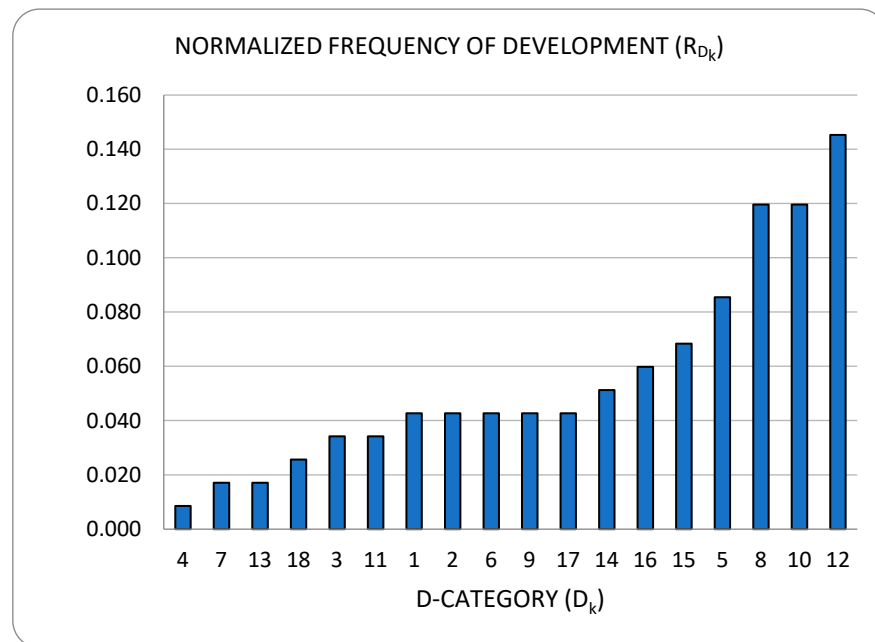
Figure 22. Normalized frequency of Developments.

Table 2. Challenge category codes and representative challenge questions.

CAT #	CODE	PROBLEM
P1	Architecture of VEs and PMs—profiling power consumption	What is the impact of specific major architectural features of computer hardware on the power consumption of VEs?
P2	Virtualization genre, platform	Is it possible to meaningfully rationalize the behavior of VE power consumption across different implementations of systems for virtualization?
P3	Estimation of power consumption of virtualization-host systems	Can host system power consumption be predicted on the basis of VE activity?
P4	Workload-adaptable power models	Can workload-adaptable power models be developed?
P5	Resource heterogeneity	How do a VE's power consumption and power model vary with resource configuration (heterogeneous VEs)?
P6	Influence of temperature on power model	How does temperature of operation affect VEs' power models?
P7	Resource use and measurement of VEs	How can load be targeted at specific VE resources? How can the actual resource consumption be measured?
P8	Host system power attribution	How can the (measured) power consumption of a host be attributed to the hosted VEs?
P9	IO Virtualization	What is the impact of specific major implementations of IO virtualization on the power consumption of VEs?
P10	Network functions	Which particular implementation of a network function is most power or energy efficient?
P11	Software L2 data plane switching	What is the impact of specific major implementations of software layer 2 data plane switching on the power consumption of VEs?
P12	Investigation of processor green capabilities	How can we model operation under real-time exploitation of processor green capabilities?
P13	Improvement of power efficiency of high-performance IO virtualization	How can we improve the power efficiency of high-performance packet IO frameworks?

Table 3. Approach category codes.

CAT #	CODE
A1	Power attribution principle
A2	Modeling bias
A3	Green operating principles
A4	Physical analysis
A5	Identification and use of metrics of energy efficiency
A6	Managed resource provision (setup time)
A7	Controlled resource provision (operation time)
A8	Resource-specific workloads
A9	Workloads representative of real use
A10	Computing resource instrumentation
A11	Workload characterization and instrumentation
A12	Direct power measurement
A13	Hardware instrumentation of predictors
A14	Simulation
A15	Use of test data in post-training phase
A16	Corroboration through experimentation
A17	Model adaptation technique

Table 4. Development category codes.

CAT #	CODE
D1	Host models -> linear regressions: predictors = computing resources
D2	Host models -> non-linear regressions: predictors = computing resources
D3	Host models -> machine-learnt: inputs = computing resources
D4	Host models -> mass storage energy consumption
D5	Adaptable VE models -> linear regressions: predictors = computing resources
D6	Adaptable VE models -> power, exponential and log regressions: predictors = computing resources
D7	Adaptable VE models -> machine-learnt: inputs = computing resources
D8	Host/VE models of power consumption -> predictors = workload characteristics
D9	Host/VE models of power consumption- > observations on operation of processor green capabilities
D10	Relative accuracy of modeling approaches
D11	Power's dependencies- > physical and logical artifacts -> component implementation
D12	Power's dependencies -> physical and logical artifacts -> architecture
D13	Power's dependencies -> physical and logical artifacts -> IT system implementation -> dependence on resource provision
D14	Power's dependencies -> physical and logical artifacts -> IT system implementation -> dependence on virtualization/containerization genre and technology
D15	Power's dependencies -> networking workloads -> software L2 data plane switching
D16	Power's dependencies- > networking workloads -> in I/O virtualization
D17	Power's dependencies- > networking workloads -> in VNFs
D18	Power's dependencies -> networking workloads -> in general networking

3. Thematic Analysis

The themes presented in this section are the product of a thematic analysis undertaken according to the method described in [1]. We first present an overview through a graphic (Figure 23) that groups the themes and then proceed to an exposition of the themes within the sub-sections.
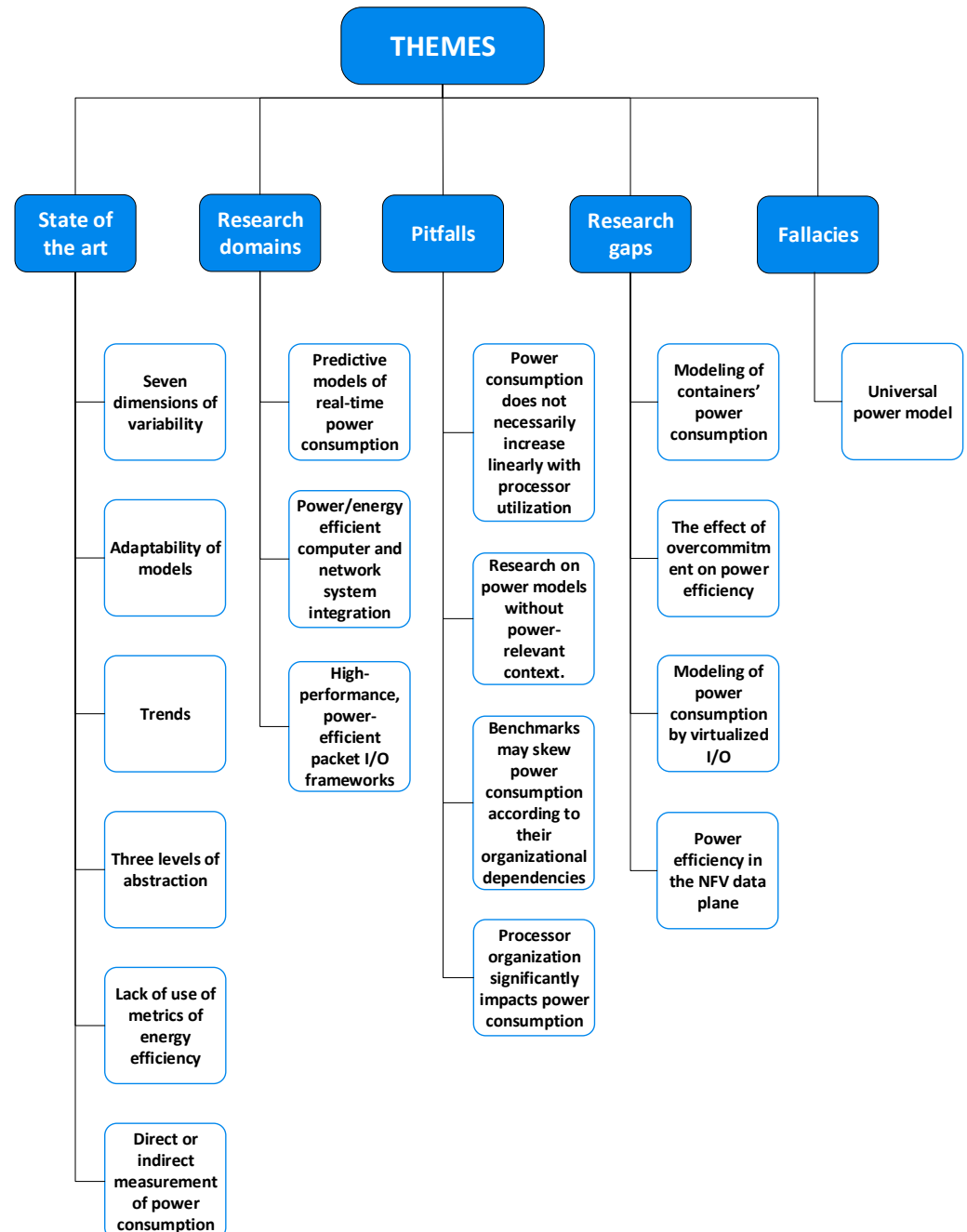
Figure 23. A graphical overview of the thematic analysis of the research space.

3.1. State of the Art

3.1.1. Seven Dimensions of Variability

Through our collation and resolution of the derivatives of the core challenge in this research space, we observe seven dimensions of variability in modeling power consumption by a virtual entity. Comparison with the power consumption of physical machines throws the core challenge into sharper relief. With physical machines:

- Power consumption can be measured directly;
- There is no virtualizing agent to consider;
- The activity of other physical machines (that do not send or receive workload) is irrelevant.

The seven dimensions of variability are shown in Table 5. It is not surprising that the problems we have seen researchers tackle are closely aligned with these dimensions. The research space is precisely about the need to obtain an understanding of the impact that these variables have on power consumption. The scope of models we found in the RUs covers only a subspace of the seven-dimensional space but the extent is not usually stated. We touch upon this issue briefly in our treatment of the pitfall of research without context and the fallacy of the universal power model.

Table 5. Seven dimensions of variability.

Dimension	Problem Category
1. Workload type	P4, P7
2. Virtualization agent	P2, P9, P11, P13
3. Host (resources and architecture)	P1, P3
4. Temperature	P6
5. Power attribution	P8
6. Co-hosted, concurrent VEs	P8, P5
7. Frequency	P12, P13

3.1.2. Adaptable Models

We observe ***three conditions*** which must be met for an automated modeling system to obtain a model of VE power consumption.

1. The first fundamental condition is common to all successful research into modeling of VE power consumption. ***Resources utilized by the VE*** (whether measured through architectural or microarchitectural instrumentation) ***must be strongly correlated to power consumption by the VE as well as power consumption in host system overheads***. If the overheads are uncorrelated to the VE's activity, or weakly so, then any significant power overhead must be modelled through a separate consideration of its driving causes, e.g., temperature [81].

Furthermore, in order that modeling may apply across a diversity of operating contexts, two other conditions must hold.

1. Any change in the parameters of correlation must be dynamically learnt and the model adjusted.
2. Any change in operating context that invalidates the parameters of correlation must be of finite duration. An indefinite transient precludes the formation of a model.

To this observation on the three conditions, we add another observation. Two of the seven dimensions of variability are commonly investigated in terms of the validation of the accuracy of modeling systems: ***workload type and co-hosted, concurrent VEs***. We suggest that a modeling system may be labelled adaptable if, minimally, ***it meets the above three conditions under these two dimensions, i.e., (a) operations of variable workload type per VE, and (b) a variable number of concurrent, co-hosted VEs.*** Before proceeding to refer to validations observed in the RUs, it is useful to draw attention here to the need for a ***modeling system***, rather than simply a model, in the estimation of power consumption. Dynamic adjustments are affected through the intervention of such a system, which adjusts model parameters to the operating context. We now proceed to describe how the variables in the two dimensions were varied in some studies.

Workload type: Different types of workload correspond to different use of resources; hence, the behavior of power consumption also differs. Several researchers [48,75,76] studied the effect of changing workload type either by categorizing the workload itself or

by categorizing the resources consumed by a specific kind of workload. Both [75] and [76] characterized the use of CPU utilization for workloads, which is known to have a workload-type dependent relationship with power consumption [62]. Thus, in both these studies, the need arises to re-train on the change of workload type (second condition). Further, both carry an observation about the duration of re-training for refined models (third condition).

On the other hand, each one of [62,69,79,81,82] is capable of adapting to different workload types without re-training. All predictors here are event counters. However, events are not necessarily linear predictors of power consumption across all workload types. In [84], it was found that the model parameters of a linear regression of event counters onto power consumption are workload specific. Given the contrast with [73,74], it seems that the root cause is the selection of events for prediction.

Concurrent operation: In [61], a non-linear model of the dynamic power of a multi-(virtual)-core VM is obtained. The dynamic power p_{vm} is expressed in terms of the average utilization u_v which n virtual cores impose on a total of N physical cores:

$$p_{vm}(u_v, n) = \alpha \left(\frac{n}{N} \right)^{\beta} u_v^{\beta}$$

Parameters α and β are determined through linear regression of the logarithmic form. The relationship was successfully tested under the operating context of one VM and three VMs. The model is limited to processor-intensive workloads; indeed, this is not surprising, since the predictor is (virtual) processor utilization. However, a conclusion can be drawn: the accuracy of the model in predicting the dynamic power of an individual VM is independent of the concurrent operation of other VMs (albeit for a limited range of workload types).

In [75], it is shown that estimation based on processor utilization can accurately predict the power consumption of multiple concurrent VEs. The prerequisite is that a model for that VE's workload type has been learnt. The prediction accuracy for individual VEs is independent of concurrent operation. The same conclusion can be drawn for [61]; furthermore, here the range of workload types has expanded to a broader range. With regard to the results obtained in [84], the conclusion is yet again the same: individual prediction accuracy (albeit workload-type-dependent) is independent of concurrent operation.

Variability in both workload type and number of concurrent, co-hosted VEs is manifestly claimed in [62,69,81]. It is noteworthy, therefore, that not only is the modeling system adaptable, but it also produces a model that is itself adaptable without the need for real-time adjustment.

We conclude this sub-sub-section with the observation that limited adaptability (defined as independence of workload type and number of concurrent, co-hosted VEs) has been achieved. From the surveyed RUs, we conclude that, presently, the limits of adaptability are the following:

1. Workload type: the power consumption of processor-intensive, memory-intensive, disk-intensive workloads and mixes thereof has been modeled by a single modeling system in an automated manner;
2. Co-hosted, concurrent VEs have been modeled up to but not exceeding over-commitment of processor physical cores.

Caution has been exercised in claiming these limits. For example, while the commitment of logical cores (e.g., Intel Hyper-Threading logical cores) has been investigated, we have found no evidence that power consumption has been modeled in a manner that automatically adapts to a transition of consumption from physical to logical cores.

3.1.3. Lack of Use of Metrics of Energy Efficiency and Standards to Address This Deficiency

We note several experiments, e.g., [55,56,68,71,103–105], that target power consumption but less than 12% of the RUs in our corpus approach the problem in terms of some energy efficiency metric [49,50,65,85]. It is necessary to move beyond measurements of how much power was consumed, to measurements of how much power was consumed to

carry out a specific task. This change in approach facilitates comparison between research works. More importantly, it directly addresses the question about the cost of operation of infrastructure.

This approach requires identification of a unit of comparison that transcends the boundaries of disparate systems that deliver this unit. This unit of comparison is referred to in the Life Cycle Assessment (LCA) framework (ISO 14040) [106] as the *functional unit*. A definition specific to telecommunications equipment is given in [107]: the functional unit is defined as "a performance representation of the system under analysis". This definition is too broad; therefore, units specific to a variety of classes of equipment are defined too [107]. In our corpus, two approaches we have seen are hash/J [50] and J/Web Interaction [65]. The functional units in these cases are the performance of one crypto-hash and one web interaction, respectively. Another is to define a functional unit specific to a digital service delivered over a telecommunications network, e.g., ten minutes' time of browsing [108]. Note that L.1310 [107] recognizes both metrics where energy is in the numerator [65] as well as those where it is in the denominator [50]. Guidance is available: energy efficiency measurement for several NFV components has been standardized, as well as measurement standards for servers, switches and virtualization systems [109].

3.1.4. Trends

Here, we present trends which we have identified through our collation of problems, approaches and developments. This sub-sub-section is divided into four parts, regarding trends in: challenges, complexity of tackled problems, approaches and developments. Within each part, trends are numbered using Arabic numerals, to differentiate cleanly between them.

Challenges

1. The causality DAG (Figure 18) shows that the research space can be characterized succinctly, note the thickness of the links originating at P7 (VE resource use and measurement) and P8 (how to attribute host system power to VEs).

    a. Figures 1 and 20 show that the challenge-category tackled most frequently ($R_{P_7} = 25.3\%$) is how to load the VE's resources and measure the loading (P7). P7 also has the most frequent presence in RUs: $F_{P_7} = 65.7\%$) (Figure 20).

    b. The accurate specification and measurement of load is essential to model formation. These measurements provide the aggregated (i.e., indiscriminate of which VE is consuming) predictors (the predictors are the input side of the model). Disaggregating the predictors and attributing measured power consumption to the individual VEs (the output side of the model) constitutes the second most frequently tackled challenge ($R_{P_8} = 16.5\%$). P8 also has the second most frequent presence in RUs: $F_{P_8} = 42.9\%$.

    c. In the following, the vector (P, A, D, weight) will be used to indicate a path (P, A, D) and the number of triads (weight) through the path:

        i. A total of 34.2% of triads collected regard P7 and P8, respectively. The triads graphic (Figure 19) shows that efforts rooted in these two challenges converge on a common objective: building adaptable models of VEs' power consumption, notably using regressions to linear combinations of computing resource predictors;

        ii. The set of triads leading to linear regressions consists of {(7,10,5,20), (8,17,5,13), (7,9,5,12), (8,1,5,10), (8,2,5,8), (7,8,5,6), (8,12,5,4) and (7,6,5,3)} and these account for 11.3%.

2. The third most frequently tackled challenge is that of the estimation of the virtualization-host-system power consumption ($R_{P_8} = 15.4\%$). This category might be overlooked in a first inspection of the research space, as it might seem futile to attempt to estimate a power consumption which can be measured. However, in practice, the logistical

challenge of the measurement of horizontally scaled system deployments seems to be well known and several works have been undertaken to develop software meters.

3. Several significant links originate on P1 (profiling power consumption's dependency on architecture). The DAG (Figure 18) indicates that approaches to tackling P1 are spread across a mixture of managing resource provisions, the use of synthetic (resource-specific) workloads and predictions using system software's instrumentation.

4. Another large group of links originates on P3 (estimation of host system power consumption). The DAG (Figure 18) indicates that a primary concern in tackling P3 is the type of model to select. The most common choices are linear and non-linear; machine-learning techniques are the least common of the three. The thickness of the triad (3,2,10,11) (D10: relative accuracy of formal approaches) indicates that there is already significant interest in whether the advanced models are worth the effort to develop them and the computational resources required to operate them.

5. While $F_{P_7} = 65.7\%$ (loading VEs and measuring their use of resources), only 8.6% ($F_{P_{7.2}} = 8.6\%$) of all RUs investigate loading containers and measuring their use of resources. With virtual data plane devices, the figure is even lower: $F_{P_{7.3}} = 8.6\%$. This imbalance suggests that there is much room left for research into modeling power consumption by containers and data plane devices.

Challenge complexity

In [1], we suggest that the diversity of approaches is a metric of the complexity of the challenge. We note (Figure 20) that challenge complexity (W_{P_k}) generally follows the frequency with which a particular challenge-category is tackled. That is: the more frequently the challenge is addressed, the more diverse are the approaches applied to it. This can be verified by noting that the heights of both sets of bars in the chart (superimposed on the same graphic) follow roughly the same pattern. However, some categories do stand out. For example:

1. P6 regards the influence of temperature on the power model. Approach diversity is poor here because only the use of additional instrumentation can be attributed to this challenge. Model bias might be attributed to this challenge too but, largely, model bias is determined by other challenges within the scope of the RU.

2. While host system power attribution (P8) has the second highest research interest (and frequency of occurrence, F_{P_k}), the number of approaches taken to solve this challenge is relatively small.

3. On the other hand, power consumption's dependencies are tackled by a disproportionately large number of approaches. This is not surprising, as the objects of study (architecture, virtualization platform, virtualization genre) are multi-faceted and dependencies can be investigated through a variety of approaches.

Approaches

Our approach utility metric, U_{A_k} [1] (Figure 21), seems to be a useful one. It communicates clearly what we have observed in our surveying. Below, we draw attention to saliencies perceived during surveying and confirmed by the metrics.

1. Instrumentation of the consumption of computing resources (A10—which includes microarchitectural instrumentation and that abstracted by system software) is repeatedly adopted ($U_{A_{10}} = 15.5\%$, $R_{A_{10}} = 15.7\%$) in empirical work in this field. It is also the most utilized of all approaches. In comparison, workload instrumentation accounts for 8.2% utilization ($R_{A_{11}} = 5.2\%$).

2. Resource-specific workloads (A8) are the more utilized approach to loading VEs (14.3% of all triads). This approach category is the second most utilized. Workloads representative of real use (A9) account for 9.0% of all triads. The corresponding frequency of occurrence figures (R_{A_k}) are 13.4% and 10.0%, respectively.

3. Two other high-utility approaches are (a) modeling bias (A2, with $U_{A_2} = 11.9\%$, $R_{A_2} = 11.6\%$) and (b) the managed provision of resources (A6, with $U_{A_6} = 10.2\%$,

$R_{A_6} = 10.8\%$). The use of representative workloads (A9) follows at $U_{A_9} = 9\%$, $R_{A_9} = 10.1\%$.

4. A comparison of the patterns of bar height distribution for U_{A_k} and R_{A_k} reveals that some categories stand out.

   a. While A12 (direct power measurement) is not as frequent as the other software-based forms of instrumentation, this approach has a utility that sticks out of the pattern (Figure 21). The reason is that most, if not all, developments obtained in an RU that include this approach depend on the directly measured power.

   b. Similarly, while workload characterization and instrumentation (A11) are employed with a frequency that is about one-third that of its alternative (i.e., computing resource instrumentation, A10), it has a far better utility-to-frequency ratio than A10. The triads graphic (Figure 19) indicates that one important cause of this high utility is that software-L2-data plane switches and (virtual) network functions are investigated primarily using measurement of workload (and not measurement of computing resources consumed to process a workload).

Developments

1. A total of 55.6% of all identified developments are obtained in modeling power consumption. The remaining 44.4% regard how power consumption depends on implementations. The implementations investigated (for their impact on power consumption) range from entire virtualization platforms (e.g., KVM) to components (e.g., processors).

2. The most frequent ($R_{D_{12}} = 14.5\%$) category of developments is that which regards observations on the dependency of power consumption on architecture. The cause of this high frequency is that developments in the study of architecture establish directional (negative, positive, neutral) correlations rather than predictive forms. For example, in a single RU [67], all of the following developments emerge:

   a. D12.08: VM power consumption increases linearly with a vCPU frequency of operation when the vCPU is operating at 100% utilization;

   b. D12.09: Virtualization-host-system power consumption increases linearly with the number of physical cores operating at 100% utilization;

   c. D12.1: Virtualization-host-system power consumption increases with the number of VMs active on the same core.

3. Amongst developments of models, the most common category (D8) is of the type where power consumption (host or VE) is predicted in terms of workload characteristics ($R_{D_8} = 12.0\%$). The next most common category regards the prediction of power consumption as a linear function of computing resources ($R_{D_5} = 8.5\%$)). Machine-learnt models of VE power consumption comprise the second least frequent category, with $R_{D_7} = 1.7\%$.

4. Despite the frequency of developments in category D10 (Figure 22), few works [60,62,63,77,79] compare the accuracy of model types. Models have largely been treated as a means to an end, with little investigation of their relative accuracy and range of validity (in the seven dimensions of variability). This may be a reflection of researchers' interests. As the popularity of methods from the body of knowledge of data science increases, works (e.g., [63]) that span a broader range of model types may be expected to increase in concert.

5. Adaptable, non-linear VE models (D2) occupy a modest 4.3%. Such a distribution adds weight to the observation that this research space is ripe for exploration using advanced modeling techniques. Indeed, one attraction for data scientists is the relative ease with which data can be collected. However, polynomial and other types of regression to closed-form are problematic as suspicion of over-fitting increases with the order of the polynomial. For example, in [76], a sixth-order polynomial is suggested

to model the relationship between processor utilization and power consumption by a host.

3.1.5. Three Levels of Abstraction

We note that existing power models may be classified into one of three levels of abstraction. In ascending order of abstraction, these are:

- Microarchitecture and architecture;
- Simple characterization of workload;
- Complex characterization of workload.

The meaning of "abstraction" specific to our use here is perhaps most easily grasped by referring to the variables used as model inputs. In all cases, the variables are some measure of load. At the lowest level, inputs that quantify the operation of processor sub-units are used (event counters and event timers). The highest level uses inputs that quantify the demand for a telecommunications system or service. Clearly, the latter inputs are far more detached from the underlying, concrete implementation than the former.

With regard to the lowest level of abstraction, many power models use hardware resource consumption to estimate the power consumed by virtual components. One survey [105] (while comparing the available power models for processors, VMs and servers) observes that most of the power models for virtual machines use physical machine counters to estimate the corresponding resource utilization by the virtual components. We offer further insight on this matter. We concur in the observation that much current research is concerned with modeling power consumption of virtual machines. The approach may be succinctly described as estimates obtained from models trained out of either architectural or microarchitectural instrumentation data. Note that we distinguish between architecture and microarchitecture using the classical interpretation [110]. Now, the term "architecture" is severely overloaded and its interpretation can easily differ from that which we wish to exploit. In the following description of the levels of abstraction, we use the alternative "system software's instrumentation" to convey the same meaning as "architectural instrumentation" with less ambiguity.

Microarchitectural instrumentation is the lowest level of instrumentation. Power consumption is expressed in terms of variables that are defined at sub-CPU and sub-subsystem levels. The granularity of this level holds the greatest potential for accuracy, but the rate of change of the observed variables has deterred several researchers from pursuing this approach to instrumentation, citing concerns about communicational and computational overhead. This concern has been dismissed by three groups of researchers [70,86,98] who have indicated that acceptable accuracy can be achieved with negligible overheads. Given the proliferation of works based on this approach and the availability of high-level (programming) language code (HLL) that facilitates the use and the potential for the capture of physical behaviors, then ***our general recommendation is a preference to investigate the use of microarchitecture instrumentation***.

System software's instrumentation regards a class of instrumentation that has meaning across the spectrum of computer systems. The input variables, such as the CPU or network interface utilization, are produced by some digest (function) of intermediary system software. For example, an operating system's (OS) measurement of core utilization can roughly be described as the core's duty cycle on behalf of the OS. This is significantly removed from knowledge of activity within the core. While models at this level are more abstract, they are still low-level, especially when compared with the other levels.

Simple characterization of workload is a less frequently encountered abstraction, used by Enokido, Takizawa and various others with whom they have co-published [68,71,111]. These models describe power consumption in terms of fundamental descriptors of workload, e.g., the number of processes and transmit/receive data rate.

Complex characterization of workload is the least granular of the models in the survey [34,112]. The objective here is to quickly proceed to a good estimate of the power or energy required to produce the workload. This kind of model has no use in real-time control

but it is useful for macroscopic comparisons, i.e., comparisons between two disparate systems for the provision of a service. The comparison might regard two different paradigms of provision of the service, e.g., classical vs virtualized implementations. Thus in [34], the implied (system) metric is the amount of power required to deliver 1 million packets per second of throughput through an evolved packet-core's (EPC) serving gateway (SGW). In [112–114], the objective is to minimize the amount of consumed power by virtualizing baseband processing functions, evolved packet core, customer premises equipment and radio access network functions.

3.1.6. Service Determinism: A Criterion Particular to the Telco Cloud

The NFV data plane demands determinism [37,96,115]. Strictly, *service determinism* is sought since the packet arrival process is generally unconstrained. The root cause of this need is to correctly size equipment resources to meet load demands, whether throughput, latency or jitter. *This need is intense*, as it impinges on a PTNO's obligations, specified in legally binding service-level agreements (SLAs). Service determinism has been approached through the augmentation of GP hardware, with domain-specific architectures (DSAs, which we first referred to while describing problem category P1). Hardware-centric DSAs for the NFV data plane are constructed out of elements (or systems thereof) that can be divided into three groups.

1. **IO system architecture:** This group comprises the abstraction, *directly at a peripheral interface*, of functionality that facilitates the virtualization of hardware, e.g., SR-IOV (used in PCI—Peripheral Component Interconnect) and N-port ID virtualization (Fibre Channel).
2. **Processor architecture:** This comprises architectural change that facilitates the partitioning of processor resources, such as multi-core processors and NUMA.
3. **Co-processing:** Compression and decompression, encryption and decryption and packet header processing are examples of high-volume tasks that can be offloaded to co-processing subsystems. Examples include Intel QuickAssist Technology (QAT) and TCP Offload Engine (TOE).

We further observe a set of approaches that are complementary to DSAs in the quest for determinism. These include:

1. **Large memory pages:** exploitation of the facility to organize virtual memory into larger pages than the general-purpose 4 KiB;
2. **User space programming:** diversion of the control of hardware resources away from the multi-service kernel, to single-common-use user space programs, e.g., DPDK and AF_XDP.

Notably, resource partitioning (see processor architecture, above, i.e., the second hardware-centric approach), combined with user space programming (the second complementary approach), realize the run-to-completion scheduling model [96].

We conclude with an observation on *the cost of the current realizations of service determinism*. The (aforementioned) combination of core partitioning and user space programming has been widely adopted through Intel's popularization of DPDK, via Intel's open-source liaison efforts. The multi-core processor enables service determinism through an approach that is anathema to the principles of multiprogramming: the dedication of hardware to a specific task.

3.1.7. Direct or Indirect Measurement of Power in Virtualized Environments?

We observe that most research in modeling power consumption seeks to obviate the need for direct measurement through indirect measurement. This indirection consists of a measurement of resource use which has a discoverable relationship with power consumption by the entity hosting the resources. Modeling, here, has the objective of indirect measurement of a variable that is not directly accessible (power consumption by VEs), through others which have convenient and reliable instrumentation. The accessible

variables are referred to as ***power proxies***. The RAPL interface provides a unique approach to measurement as it directly addresses power consumption. However, notwithstanding appearances of direct measurement, RAPL is actually based on a software model that uses performance-monitoring counters (PMCs) as predictor variables to measure power consumption. It is available in processors starting from the Sandy Bridge microarchitecture. RAPL measures the power consumption of different physical domains, where each domain consists of either cores, sockets, caches or GPU. We briefly comment on its accuracy through references to research that has investigated them.

1. In [98] the advantages and drawbacks of using RAPL were investigated. Different Intel architectures such as Sandy Bridge, Haswell and Skylake were used in the experiments to analyze RAPL's accuracy and overhead. Data collected were modeled using a linear model and a Generalized Additive Model (GAM). The accuracy of the predicted results was compared with the measured power consumption from a precise external hardware power meter where RAPL based models show 1.8–4.3% of error for the various architectures. The prediction accuracy of RAPL-based power models was also compared with those based on OS counters, where OS-based models show a higher error of 5–16%. Furthermore, the performance overhead (in terms of time) of using RAPL was studied at different sampling frequencies and for different application runs. Results show that even with a high sampling frequency of 1100 Hz, RAPL incurs overhead of not more than 2%. Some limitations of using RAPL include poor driver support to read energy counters, the overflow of registers due to their 32-bit size and the measurement of energy consumed by individual cores.

2. Another study to analyze the precision of RAPL is presented in [116], where only the dynamic change in power consumption is observed. An external power measurement unit (WattsUp Pro) is used as a reference for power measurement values. Intel Haswell and Skylake servers were used in the experiments to run different applications and to find the reliability of RAPL with the help of external power meters. However, in this research work, only two power domain packages (power consumption of whole socket and DRAM domain of RAPL) were observed. Applications such as dense matrix multiplication and 2D Fast Fourier Transform were used for server power profiling. Results show that the power measurement error varies with changing applications and its workload size. For different applications the average measurement error using RAPL was in the range of 13–73% considering WattsUp power meter as the ground truth. It was concluded that with the modern multi-core parallel processing and resource contention for shared resources, there is a complex non-linear relation between performance, workload size and energy consumption. Hence it is difficult to attain low error percentage for power measurement using on-chip sensors.

3.2. Research Gaps

Three significant challenges remain unaddressed, while a fourth requires further attention:

1. The modeling of containers' power consumption;
2. The effect of overcommitment on power efficiency;
3. The investigation and classification of DPDK applications;
4. The fourth challenge, which is starting to receive some attention [83,85,87,94], regards the modeling of power consumption by virtualized I/O.

Note that our treatment of research gaps does not address improvements in approaches. As we have already indicated, it is in our treatment of developments that more adaptable modeling methods are required to tackle the dimensionality of the field. Similarly, we do not include the lack of use of metrics of energy efficiency with research gaps, as it is a deficiency in the approaches, not a challenge in itself. Rather, here our attention focuses on where the more pressing challenges lie for the development of power and energy control of VEs.

Gap #1: Modeling of containers' power consumption. Few works [78,81,82] tackle containers from the perspective of their power consumption. However, at least for the

telco cloud, VMs are no longer the destination (see, for example, [40,117]. Containers have replaced virtual machines as the base for deployment of virtual network functions. In [78], the approach(-set) taken is to:

1. Use representative workloads, e.g., HEVC (High Efficiency Video Coding) transcoding and machine learning image classification;
2. Customize the set of low-level instruments used to correlate power and energy consumption with workload characteristic.

This work develops a profiling tool. It provides guidance that is specific to the application and both the hardware and software aspects of the containing platform. "[P]olicy" for "tradeoff between energy, power and application performance" is the cited objective. Given the high-dimensionality of the core challenge, this set of approaches to the modeling and measurement of power consumption, may well be more coherent with the European Telecommunications Standards Institute's (ETSI) Management and Orchestration (MANO) standard. Such information would then be included in the infrastructure-resource-requirement's meta-data descriptors in the VNF package [118].

Gap #2: The effect of overcommitment on power efficiency. Overcommitment consists of the allocation of more capacity of some compute resource to VEs, than is physically installed. The concept is very similar to the oversubscription of telecommunications capacity to subscribers, such as when the arithmetic sum of capacities of access links exceeds the aggregating device's backhauling capacity to a central office/local exchange. In this context, overcommitment principally concerns processing cycles and memory space. As with oversubscription, there is an optimization problem to solve. One problem of interest to this survey's scope is understanding the relationship (say, ratio) of committed virtual resources to installed physical resources that optimizes the total cost of ownership (TCO) of cloud infrastructure:

- On the one hand, the facility to overcommit has a direct impact on the density of the packing of VEs (number of concurrently active VEs) on a virtualization-host-system, thereby reducing the TCO;
- On the other hand, overcommitment may reduce the power efficiency of a workload.

Had this challenge been tackled in any depth, or at least in any breadth, it would have merited a category of its own. Currently, however, we are only aware of a single study [65] that tackled this challenge. The results obtained strongly justify overcommitment of processor cores to vCPUs, for the case of transaction web service workloads, with the increase in throughput (measured in web interactions per second, or WIPS) increasing at a faster rate than power consumption. This behavior was observed well into overcommitment ratios of processor cores to vCPUs equal to 3 (three). The overcommitment of physical to VM memory was not investigated.

Gap #3: Investigation and classification of DPDK applications. In Section 3.3.2, we address the relationship between DPDK and power consumption. In the course of a public discussion in the forum offered by the North American Network Operators' Group (NANOG) [119], there emerged a need for clarity on DPDK's association with inefficiency in power consumption. Interest was particularly expressed in the knowledge of a classification of extant DPDK applications according to their power consumption, and contribution to the code base to improve power-hungry applications. In summary, among high-performance packet I/O frameworks, a classification of applications that use DPDK APIs in order to assess their power consumption and correct usage would be of particular interest, given their diffusion and potential.

Gap #4: Modeling of power consumption by virtualized I/O. Power consumption of network I/O has been investigated to some extent as this is central to the feasibility of network functions decoupled from hardware. While software and hardware solutions are already available (see Section 3.3.2), they require frequency and idling control targeted to their specific operating conditions. Notably, naive DPDK runs the processor core at its maximum power consumption, regardless of load. The exploitation of adaptive-rate

(AR) processing and low-power idle (LPI) should provide a means to save power while processing high networking loads. However, effective control of these means is still elusive, despite both using Xeon Haswell microarchitectures, refs. [83,87] reach opposite conclusions about the feasibility of processor core C-states. The former [83] finds LPI an effective means of reducing power consumption of packet forwarding (with limitation on latency) while the latter [87] finds it ineffective, preferring the use of the **pause** instruction. Furthermore, in [120], performance state transitions (P-state) are found to impose a high transition latency, while in [87], P-state regulation is the preferred approach. There is scope for research in the dynamic adaptation of the processor's operating state to save power.

3.3. Pitfalls

3.3.1. Power Consumption Does Not in General Increase Linearly with Processor Utilization

Notwithstanding advances made in identifying operating contexts that manifest a sub-linear power-utilization relationship [62,69], more recent publications [57,121–128] persist in using the linear model without acknowledging its limitations. The model is simple to use and has some foundations in research [129]. It has three premises, described here with regard to the operation of Microsoft Windows:

1. When Windows has no threads to run on a logical core, it schedules the idle thread [130].
2. The idle thread keeps the processor in a low-power state [131]. The specific state depends on the processor's green capabilities.
3. In the complement (non-idle time), the processor issues instructions at a constant rate.

This simple model has limitations [62,132]. It fails to take into account diverse processor operating contexts, some of which are coming to bear on current use cases. Specifically, the third premise is true only to the extent to which instructions are being fetched and data are being loaded from/stored to an instruction and data cache, respectively. Consider the context of 90% and greater hit ratios. At such cache hit ratios, the rate of instruction issue is expected to be narrowly distributed about its mean. By contrast, the lower the hit ratio at the cache level before main memory, the lower the fraction of non-idle time at which power consumption saturates. This saturation is strikingly illustrated in [62]. The variation of power consumption due to the execution of tests from the SPEC CPU2000 benchmark suite is shown. The power consumption diverges at 25% CPU utilization and the consumption of the processor-bound test (mesa) is greater than that of the memory-bound test (mcf) ***by a factor of about 2.6***.

Another good (albeit broad) illustration of this pitfall is given in [133]. Data on power consumption and CPU utilization under a standardized benchmark are plotted for four different physical server models. None of the relationships are linear. Neither is there a single, common behavior.

Researchers align themselves into two groups with regard to CPU utilization. One group favors (operating-)system metrics (of which CPU utilization is one metric) and the other favors event counters (microarchitectural instrumentation). The arguments posed by each group against the other's approach can be summarized as follows. The "system metrics" group claims that the "event counters" group's work is (a) not portable (at least across microarchitecture families) and (b) cannot be exercised without low-level access to the host (therefore, this approach cannot be exploited by user-level privileges) (see, e.g., [134] and [75]). The "event counters" group claims that CPU utilization is a workload-dependent predictor (see, e.g., [59]) and therefore cannot be used without re-training the model. Indeed, this modification to the "system-counters" approach is employed in [76], where it is stated that "[b]ecause of changes of VM's internal applications ... parameters must [be] recalculated automatically". ***Given these arguments, it seems that the system metrics group argument is weak: both system metrics and event counters require re-training if hardware is changed but system metrics lack the granularity to discriminate between workloads*** (cf. [62]). This means that CPU utilization can only be used as the sole

predictor if it is re-trained with a change in workload. We deal with this problem, which we have termed the fallacy of the universal model, in Section 3.4.1.

With regards to the use of hardware threads (Intel®Hyper-Threading), we have observed that various works concur on the operating context under which a linear relationship is subject to the lowest error. This includes at least the following two conditions:

1. The processor cores are increasing their instruction issue rate in proportion to the fraction of time they spend busy. This implies that instruction and data cache hit ratios are high. This is simply the third premise;
2. Only one logical core is active per physical core at any given time [62,63,69]. Expressed alternatively, actual utilization must lie below half maximum utilization. The underlying cause is that activation of the second logical core employs fewer organizational units of the processor than activation of the first logical core.

The first condition is particularly problematic, as cache miss ratios are likely to be much higher in the context of virtualized environments. In such environments, the number of runnable threads is the sum of runnable threads controlled by independent operating systems. Evidently, this is higher than the expected number of runnable threads on a single server instance.

Other evidence of this "utilization trap" is not hard to find. In [57], the compute resource is stressed using cpulimit and stress-ng. The "cpulimit" utility runs a specified process image, then pauses and resumes it until a certain percentage utilization is reached [127]. The repetitive execution of a single process is highly likely to create conditions for very high instruction- and data-cache hit ratios. Such favorable hit ratios skew results towards the linear relationship between CPU utilization and power consumption.

3.3.2. DPDK Is Not Intrinsically Inefficient in Power Consumption

Research on power efficiency in DPDK applications [48,51,53,87] has portrayed DPDK as power inefficient. Before proceeding to our exposition of this pitfall, we distinguish between data, control and management planes. "Data plane" is a term used to refer to the infrastructural means that provide the capacity for exchange of customer (or subscriber, or end-user) data. It is complemented by a control plane, which refers to those means that facilitate the dynamic setup, maintenance and tear-down of a functional data plane. Another complementary part is the management plane. This includes the infrastructural means for a network operator to configure and monitor the control plane and the data plane, as well as intervene to correct faults arising in either plane. Simpler networks may have no control plane.

Now, we proceed to the exposition of the pitfall. In one particular case [48], it is claimed that "we found that a poll mode driver (PMD) thread accounted for approximately 99.7 percent CPU occupancy (a full core utilization)." The implication that seems to emerge here is that the PMD itself is driving this power consumption.

This portrayal is problematic at best and incorrect at worst. The referenced investigations of DPDK have indicated a very low power efficiency, **but they do not clearly distinguish between the responsibility of the DPDK API and the application using it** (the API). A recent, public thread [119] has emphasized the responsibility of the application developer in the avoidance of the naïve, "default approach" of busy polling. Such an approach would, indeed, poll network IO hardware continuously [48], truly fitting the epithet "spinning-hot" [37]. However, a broader (in the sense of including industrial correspondents) investigation [119] suggests that:

- Contrary to claims in [48], it is the driving behavior of OvS that is inefficient in power consumption;
- There are simpler, technical means of throttling a polling loop, including, say, the use of program code to interleave ACPI C1 states with polls according to traffic demands.

These latter observations cast doubt on the claim that automated frequency control is outside the scope of current frequency governors, since "the OS won't be able to distin-

guish whether it's under a heavy load" [48]. On the other hand, savings through NUMA awareness [51] (where transmit/receive port, memory and processor core are kept within the same NUMA node) is affirmed in [119].

3.3.3. Research on Power Models without Power-Relevant Context

This pitfall traps readers who attempt to draw conclusions from published research which lacks a clear specification of context relevant to power consumption. The pitfall is best illustrated through examples.

1. **Failure to emphasize context: idle power consumption vs frequency**. The dependence of idle power consumption on clock frequency is context sensitive. In [49], it is explicitly stated "idle power consumption remains constant, regardless of the CPU frequency . . . across the whole frequency range" (1.6–2.6 GHz). The CPU is an Intel Xeon E5620. In [81,82], a quadratic relationship between idle power consumption and frequency is observed. Here, the CPU is an Intel Core i5 Haswell. In these two instances, emphasizing the restricted scope of findings would suffice to spare a reader from excessively broad inferences.

2. **Failure to emphasize context: idle power consumption vs hardware and software specification**. Enokido's and Takizawa's work [71] derives a power consumption model for a server while VMs run computation-bound processes. The servers used run on Intel Core i5-3230M processors. These processors are used in the mobile device market [135]. They are capable of low-power idle states [136]. CentOS 6.5 uses a tickless kernel [137]. Combined, these facts, relevant to the context of power consumption, provide a plausible explanation for the observed increment in power (denoted, in [71], by $minC_t$), when a core in a package is activated. Again, therefore, the scope of findings is likely to be restricted.

3. **Failure to fully define context: Configuration of power-relevant parameters.** We use [138] as an example. No reference is made to whether Hyper-Threading is enabled. This is essential to understanding how the ESXi vCPUs are created. Neither is any information given about how the vCPUs are related to physical (or logical) cores. Nor are we told how virtual network interfaces and switching are set up. ESXi version 5 offers both paravirtualization ("vmxnet") and emulation ("e1000") to implement virtual network interfaces. The impact on energy consumption of selecting a virtual network interface implemented by emulation can be expected to be high [56].

The examples cited illustrate the importance for a researcher of power models to qualify his/her results *with a well-defined physical context*. Research into power models involves hard components and a diligent characterization thereof is essential to the acceptance of work as scientific research.

3.3.4. Benchmarks May Skew Power Consumption According to Their Organizational Dependencies

We have seen that both "cpulimit" and "stress-ng" do not produce generally representative measurements of power consumption. This observation is not limited to the measurement of power consumption. The use of kernels, toy programs and synthetic benchmarks to measure performance has been identified as unrepresentative [110] of general performance. Benchmarks are standardized workload generators that are used for the comparison of computer systems for a specific class of application. Unless this application class is a good representative of the application of the computer system in productive use, the power consumption measured under test is not a reliable predictor of that obtained during productive use. It is necessary to plan test workload generators in advance and state the limits of the validity of results. In [65], TPC-W is used, which is a transactional web benchmark that can simulate the business-oriented online web servers. The MySQL++ Java version of TPC-W benchmark, suitable for cloud applications, is used to generate the online traffic, where three different traffic profiles based on the browsing, purchasing and

ordering of books are generated. The throughput measure for these servers is observed through the metric Web Interactions Per Second (WIPS).

3.3.5. Processor Organization Significantly Impacts Power Consumption

We illustrate this point with a wide-ranging example [139] which compares the Intel Xeon X5670 and AMD Opteron 2435.

1. Different idle loops (using no operation, pause, repetition, etc.) were tested to see their effect on power consumption of both systems. It was observed that the Intel Xeon has a loop stream detector, which disables the processor's features such as fetch and decode. On the other hand, the AMD processor has no hint to process these loops efficiently; hence, it consumed more power than the Intel processor.
2. A processor consumes a different amount of power depending upon the instruction (such as load, addition, multiplication, etc.) and the level in the memory hierarchy which is accessed by the instruction.
   a. For the Xeon, data throughput of all instructions from a particular memory hierarchy level is almost the same, but there is a difference in their power consumption. The 'load' operation consumes the lowest power compared with other instructions, and this holds true for all memory hierarchy levels. The reason is that the 'load' instruction just needs to load the content on the processor registers whereas 'add' and 'mul' operations are more computationally demanding.
   b. However, the AMD processor's behavior is the opposite. When the 'load' operation accesses the L1 cache, it achieves almost one-and-a-half times the data throughput of other operations and hence also consumes more power. This difference in resource utilization is due to the different microarchitecture of AMD processors, where the 'load' instruction is handled by many floating-point pipelines. Other instructions just use a single pipeline for their operations. Moreover, AMD processors have an exclusive cache level design, which requires write-back when evicting data among different cache levels. Since Intel's inclusive cache design does not require this function, it consumes less power. Within higher memory hierarchy levels (L2 or L3 or main memory), the AMD's computation ('add' and 'mul') and data transfer operations ('load') deliver roughly the same data throughput and consume roughly the same power.

3.3.6. Isolation of VE for Power Modeling and Measurement

Isolation of any VE from its hardware counterparts cannot be achieved completely [140]; thus, the assumption of measuring power consumption of an individual virtual entity irrespective of the hardware on which it is implemented is an illusion. The virtual infrastructure is composed of several components at both hardware and software level, where the effect of underlying hardware, OS and VNF technology can significantly impact the power consumption. Hence, isolation as well as the modeling of power consumption for an individual virtual component is difficult to obtain.

3.4. Fallacies
3.4.1. A Universal Power Model

We have suggested that the core challenge in modeling power consumption by VEs lies in the number of dimensions of variability. This has been demonstrated throughout this survey, where a number of generalizations have been addressed. In summary, the literature shows that:

1. Host power consumption does not generally have a linear relationship with processor utilization;
2. CPU-intensive workloads that repeatedly execute the same code skew power consumption results;

3. Network-intensive workloads are power- and time-consuming because they employ emulations of network switches, but the root cause (emulation in the hypervisor software switch) disappears with SR-IOV [141];
4. Host saturation must be taken into account in predicting VEs' power consumption;
5. Processor utilization (an architectural attribute) is insufficient to predict host power consumption and microarchitectural attributes, such as LLC misses, are necessary to predict host power consumption even for the same level of processor utilization.

This list, while not exhaustive, amply illustrates that the several dimensions of variability are significant in the determination of VE power consumption. A model claiming to determine power consumption as a function of fewer variables than the dimensions we have pointed out *must be accompanied by a scoping region that limits its use*. While a precise scope may be an unrealistic demand, it is essential that guidance be given about the conditions of the use of the model. We now illustrate this point by using two examples from the corpus.

Example #1: Khan [50] compares energy efficiency (hash/J) obtained by scheduling process threads on additional cores, with that obtained by scheduling them on hardware threads on active cores (through Intel Hyper-Threading). He shows that the former is greater than the latter. In apparent contrast, Enokido and Takizawa [68] show that for a given data transmission rate through the uplink of a software virtual switch, greater energy efficiency (W/bps) is obtained by operating an additional hardware thread on an active core (through Intel Hyper-Threading), than operating an otherwise idle core. An important difference lies in the task's processing "intensity", i.e., the rate of the supply of instructions. While Khan's operations are tightly bound to the processor (cryptographic hashing), Enokido's and Takizawa's operations are distributed over the processor and network input/output. Without delving into detail, it is realistic to hypothesize that the average instructions per second demanded are far lower in the networking application, since the transmission of a large file (as is the case here) does not take place in one processing burst. The operating time is divided between the processor and the media channel. In such a scenario, the added capacity of the same-core hardware thread suffices.

Example #2: At the time of writing, the scope of validity (where the scope is a subspace of the seven-dimensional space) is typically only implicit. Notably, in [75], a "refined model" is used as a means of the accurate prediction of power consumption by virtual machines while running very specific benchmarks. It is also noteworthy that the authors contemplate a type of onboarding process wherein "new" VM entrants to a cloud are modelled as a prerequisite to their inclusion in the power-prediction system. Indeed, such a process is already intrinsic to the management and orchestration of virtual network functions. Just as the virtual deployment unit (VDU) nodes (in virtual network function descriptors (VNFDs)) store VM properties describing computer system resource demands, so can the descriptor template be extended to provide properties regarding power consumption demands. This "onboarding" is necessary since the selected predictors and modeling do not cover a sufficiently broad range of workload types, and a specific model must be learnt online, i.e., on the fly.

On the other hand, we propose that a comprehensive power model for existing implementations may be possible, under two conditions.

1. Every resource that consumes power must own a counter that registers its usage, or lack thereof, during a specific clock cycle.
2. Usage of a specific resource during a specific clock cycle must consume a constant amount of energy. This has the following corollaries:
    a. Energy consumption by the specific resource is a linear function of the number of clock cycles for which the resource is active;
    b. Power consumption of a system can be expressed as a linear combination of the total set of such resources;

    c.   The amount of energy consumption by a specific resource during a specific clock cycle must be independent of usage of other resources during any other clock cycle.

4. Conclusions

We conclude by summarizing our contributions (Section 4.1) and suggest a framework for future research into real-time, predictive models of power consumption by VEs (Section 4.2).

4.1. Contributions

We have identified seven dimensions of variability (workload type; virtualization agent; host resources and architecture; temperature; power attribution; co-hosted, concurrent VEs; and (clock) frequency of operation) and observed that the challenges tackled have aligned themselves with these dimensions. This breadth has prompted us to emphasize the fallacy of the universal power model: no single power model can cover all seven dimensions through the inclusion of variables and parameters. It is essential that prospective users of any such power model be aware of the limits of its scope. On the other hand, we have pointed out that the state-of-the-art includes adaptable modeling systems that handle variability in more than one dimension. Moreover, at least limited variability in two of the seven dimensions—workload type and concurrent operation of (multiple) VEs—is commonly validated, i.e., whether the model is truly capable of predicting power consumption under variability in workload type and the number of concurrent VEs.

PAD elicits trends in its proceedings through a sample of a corpus. In particular, the following examples are among the most noteworthy (but not the only) saliencies.

1.   The challenge category tackled most frequently is that of how to load the VE's resources and how to quantify and measure the load; disaggregating the predictors and attributing measured power consumption to the individual VEs is the second most frequently tackled.

2.   The variety of approaches that tackle a (category of) challenge is positively correlated to the frequency with which it is tackled.

3.   Instrumentation of computing resources (e.g., instrumentation of microarchitectural artifacts) is the most commonly adopted approach (towards developments), surpassing instrumentation of the workload.

4.   Resource-specific workloads (e.g., processor-specific) are the most commonly utilized, surpassing workloads representative of real use (e.g., web applications).

5.   In developments, the most commonly developed model type is that where the power consumption (of the host or VE) is predicted in terms of workload characteristics; power consumption as a linear function of computing resources is second.

6.   At the other end of the frequency range of developments, machine-learnt models comprise the second least frequent category of developed models, and adaptable, non-linear VE models are also very infrequent.

The process of parsing works and aggregating their codes is, however, only the principal ingredient in the overall progression towards the end goal: a set of themes that suitably profile the works in an area of research. Indeed, these codes and their interrelationships have elicited several research gaps, pitfalls and a fallacy, as well as evidence of the state-of-the-art and of research domains.

4.2. A Framework for Development of Real-Time, Predictive Power Models

Evolution of the research space on power consumption in virtualized environments now suggests the following framework for the further development of power models:

1.   Division of the problem into:

    a.   A modeling concern:

        i.   What components to include;

ii.   What workload(s) to consider;

iii. What state factors (temperature, frequency, performance and idle states) to account for.

b. An attribution concern, i.e., how to attribute host power to VEs.

2. Division of the approach into:

a. Microarchitectural instrumentation, based on intimate knowledge of the microarchitecture and the memory system;

b. Granular attribution based on time-division multiplexing;

c. Model selection.

3. Development of parameterized models, subject to continuing (if not continuous) optimization of the parameters under machine learning.

Abbreviations

5GPPP	5G Infrastructure Public Private Partnership
ACPI	Advanced Configuration and Power Interface
and	Automatic Distribution Network
AMD®	Advanced Micro Devices
ANN	Artificial Neural Network
API	Application Programming Interface
AR	Adaptive Rate
AF_XDP	Address Family eXpress Data Path
BMAP	Batch Markov Arrival Process
BS	Base Station
BSS	Business Support System
CAGR	Compound Annual Growth Rate
CAPEX	Capital Expenditure
CNF	Containerized Network Function
COE	Cost Of Energy
CPU	Central Processing Unit
CRM	Customer Relationship Management
CSP	Communications Service Provider
DAG	Directed Acyclic Graph
DG	Diesel Generator
DPDK	Data Plane Development Kit
DSA	Domain Specific Architecture
DVFS	Dynamic Voltage and Frequency Scaling
EPC	Evolved Packet Core
ES	Energy Storage
ETSI	European Telecommunications Standards Institute
GAM	Generalized Additive Model

GHG	Greenhouse Gas
HEVC	High Efficiency Video Coding
HW	Hardware
IDS	Intrusion Detection System
IO	Input/Output
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology
ITU	International Telecommunications Union
KVM	Kernel Virtual Machine
LCA	Life Cycle Assessment
LLC	Last-level-cache
LTE	Long Term Evolution
LTE-A	Long Term Evolution-Advanced
LPI	Low-Power Idle
LXC	Linux Containers
MANO	Management and Orchestration (ETSI)
MB	Megabyte
MBS	Macro-cell Base Station
MILP	Mixed Integer Linear Programming
MIMO	Multiple-Input-Multiple-Output
MIPS	Millions of Instructions Per Second
MOS	Metal-Oxide-Semiconductor
MEC	Multi-Access Edge Computing
ML	Machine Learning
MNO	Mobile Network Operator
MTU	Maximum Transmission Unit
NANOG	North American Network Operators Group
NFV	Network Functions Virtualization
NIC	Network Interface Card
NPC	Net Present Cost
NUMA	Non-Uniform Memory Architecture
OPEX	Operational Expenditure
OS	Operating System
OSS	Operational Support System
PAD	Problem–Approach–Development
PCI	Peripheral Component Interconnect
PDU	Power Distribution Unit
PMD	Poll Mode Driver
PTN	Public Telecommunications Network
PV	Photovoltaic
QAT	QuickAssist ®Technology
QoS	Quality of Service
RAM	Random Access Memory
RAN	Radio Access Network
RAPL	Running Average Power Limit
RE	Renewable Energy
RES	Renewable Energy Source
RRH	Remote Radio Head
RU	Research Unit
SBA	Service-based architecture
SCBS	Small Cell Base Station
SDN	Software Defined Networking
SGW	Serving Gateway
SPEC	Standard Performance Evaluation Corporation
SR-IOV	Single Root Input Output Virtualization
SVM	Support Vector Machine
TCO	Total Cost of Ownership

TCP	Transmission Control Protocol
TOE	TCP Offload Engine
TOU	Time-Of-Use
TPC-W	Transaction Processing Performance Council—Web
UA	User Association
vCPU	Virtual CPU
VDU	Virtual Deployment Unit
VE	Virtual Entity
VM	Virtual Machine
VNF	Virtual Network Function
VNI	Visual Networking Index
WIPS	Web Interactions Per Second
WSC	Warehouse Scale Computing

Appendix A. Application of PAD to the Scope of Renewable Energy Use in Radio Access Networks

In this appendix, we digress to elaborate on the application of PAD to a topical scope: research into the use of renewable energy sources (RESs) in radio access networks (RANs). Our purpose here is to shed light on the mechanics of the method within a context that complements the principal context of this work.

Nine research units (RUs) [142–150] have been mined. The motivation driving this research is presented first, followed by a tabulation of the P-, A- and D-node codes and a categorization of the codes. We do not proceed to thematic analysis, as this sample is not large enough to support it. Patterns are detected and presented as *clusters* (part of the PAD method), but enunciation of a thematic analysis requires the confidence emanating from a broader search through the corpus.

Appendix A.1. Motivation

A clear sense of motivation emerges: OPEX (energy cost) must be controlled and greenhouse gas (GHG) emissions must be reduced. The Jevons paradox (alternatively expressed in the Khazzoom–Brookes postulate) emerges again: the deployment density of 5G base stations (BSs) and 64/32-channel massive multiple-input multiple-output (MIMO) antenna arrays drives "the power consumption of a 5G BS to be 2-3 times that of an ordinary 4G BS" [144]. Other works are motivated by the need to control capital expenditure (CAPEX) [142,147,148] and understand how RES harvesters compare with conventional hydrocarbon-fueled generators, therein seeking either to reduce CAPEX or to determine the true cost of wholly autonomous operation under the condition of power supply from RES.

Motivation, therefore, may be tersely summarized as OPEX control, GHG emission control, CAPEX control and autonomous operation. Within Figure 1, motivation may be located at the center of the circle, corresponding to the core challenge driving a research area.

Appendix A.2. Tabulation and Clustering of P-, A-, and D-Node Codes

The tabulation below (Tables A1–A3) summarizes two rounds about the sample: a first round of coding (code: "terse, dense representations of a verbose articulation of a concept", see [1]) and a second round wherein a first categorization (note the prefixes, for example "RAN architecture", "Radio path", "Comparison", etc.) took place. Figures A1–A3 illustrate how the nodes have been clustered. An aspect of PAD that bears emphasis here is that the coding and clustering process is an *iterative* one: the final categorization only emerges at the end of several cycles of refinement, comprising consolidation, observation, disaggregation and re-aggregation as further similarities between the codes emerge. Hence, the categories shown in Figures A1–A3 are by no means final, but they are an essential transitory stage in the progression towards the final categorization.

Appendix A.2.1. Table of P-Node Codes

Table A1. Challenges and representative challenge questions.

Problem #	Problem Code
1 [142]	How can power supply system parameters be harvested in a convenient manner?
2 [142]	How does CAPEX of diesel fuel generators compare with that of RES plant?
3 [143]	How can the limitations of individual RES be overcome?
4 [144]	Spatio-temporal distribution of light intensity and communication demand misaligns energy supply and load.
5 [144]	Disparate ownership of RES plant and automatic distribution network (ADN) infrastructure leads to concerns with security and privacy.
6 [144]	How can participation of communications service providers (CSPs) in renewable energy (RE) harvesting be encouraged?
7 [144]	How can real-time energy scheduling be improved to avoid loss of RE, over-discharge of energy storage (ES) and balance between supply and demand?
8 [145]	How well do machine learning (ML) prediction models based on past traffic and RE harvesting support RE use and Quality of Service (QoS)?
9 [145]	Which long-term strategy of operation should be used over prediction intervals: (a) turn off/on micro-BSs to match network demand OR (b) turn off/on micro-BSs to match RE capacity?
10 [146]	How can Time-Of-Use (TOU) electricity pricing, RE production and fairness of allocation of throughput to users be integrated into a modeling scheme?
11 [147]	Which system of autonomous power supply, whether renewable or not, optimizes one or more of the following objectives: net present cost (NPC), cost of energy (COE) and GHG emissions?
12 [148]	How can we exploit multiple RES to design a RAN that minimizes grid energy consumption while serving at least 95% of users?
13 [149,150]	How can user association (UA) be exploited over a short timescale in a hybrid-powered (RES + hydrocarbons) RAN to minimize grid power consumption?
14 [150]	How can traffic and weather forecasts be exploited over a long timescale to minimize grid power consumption

Appendix A.2.2. Table of A-Node Codes

Table A2. Approaches.

Approach #	Approach Code
1 [142]	Use a wireless sensor network to measure parameters relevant to power supply.
2 [142]	Systematic survey of vendors and consultation with base station operators.
3 [143]	Broad comparison of research and industrial (technical specifications) output concerning use of RES in powering BSs.
5 [144]	Organize multiple 5G BSs carrying photovoltaic (PV) panels into an energy-aggregation system architecture.
6 [144]	Regulate the mid-term exchange of energy using contract theory.
7 [144]	Regulate the short-term use of energy by a Lyapunov optimization.
8 [145]	RAN architecture: a single service area served by one LTE-A macro-BS (MBS) and six micro-BSs providing additional capacity in the service area's hot spots.
9 [145]	Energy system architecture: PV panels + energy storage (ES − batteries) + power grid.

Table A2. *Cont.*

Approach #	Approach Code
10 [145]	Traffic model: Traffic data provided by a large Italian mobile network operator (MNO).
11 [145]	RE production data estimated using PVWATT.
12 [145]	QoS: percentage of lost traffic.
13 [146]	RAN architecture: a single service area served by one LTE-A MBS and two micro-BSs providing additional capacity in the service area's hot spots.
14 [146]	Energy system architecture: MBS uses power grid, while one micro-BS is supplied by the power grid and solar energy, and the other micro-BS is supplied by the power grid and wind energy.
15 [146]	Energy system economics: power-grid energy priced according to Time-Of-Use.
16 [146]	Game theory applied to balance opposing objectives of (a) minimization of energy cost or energy consumption and (b) maximization of user-throughput fairness.
17 [146]	Alternative energy cost optimization approaches: (a) reduction by decreasing BS energy consumption; (b) reduction through direct optimization of energy cost.
18 [146]	Simulation through bespoke software.
19 [147]	Energy system architecture: Alternative combinations: (a) PV + diesel generator (DG) + ES (b) PV + ES (c) DG + ES (d) DG.
20 [147]	Design, simulation and optimization, using HOMER; optimization objective is minimization of NPC.
21 [147]	Energy consumption model: power demand data provided by a Kuwaiti mobile network operator (MNO).
22 [147]	Energy system component selection: where relevant, mono-crystalline silicon (Mono-Si) is preferred to polycrystalline silicon (Poly-Si) and Lithium-Ion batteries is preferred over other technologies.
23 [148]	MILP formulation.
24 [148]	Traffic model: Traffic data provided by a mobile network operator (MNO), both voice and data, for a cell of 0.3 km^2, showing number of simultaneous active users at any time during a day.
25 [148]	RAN architecture: eight service areas, each served by one LTE-A MBS and four micro-BSs.
26 [148]	BS power consumption model: linear power model (developed in earlier work), with hourly load factor reflecting days of the week and of the weekend.
27 [148]	RE production data (hourly) obtained from Italian RES operator's (Terna) website.
28 [148]	Radio path: 3D map of suburban area.
29 [148]	Genetic algorithm.
30 [149]	RAN architecture: a single service area served by one 5G MBS and twenty-four small cell base stations (SCBSs), with microwave backhaul from MBS to core and SCBS to MBS.
31 [149]	Energy system architecture: MBS uses power grid, while SCBSs are supplied by RESs.
32 [149]	Random-uniform spatial distribution of low-data-rate and high-data-rate users, respectively.
33 [149]	Green heuristic: prioritize green SCBSs in the UA process.
34 [150]	RAN architecture: a single service area served by one LTE MBS and thirty-six/sixteen/four small cell base stations (SCBSs).

Table A2. *Cont.*

Approach #	Approach Code
35 [150]	Energy system architecture: MBS uses power grid, while SCBSs are supplied by wind turbines without ES.
36 [150]	Energy system architecture: MBS uses power grid, while SCBSs are supplied by PV with ES.
37 [150]	MILP formulation of problem of UA with minimal grid power consumption.
38 [150]	Short timescale control of UA through three alternative policies: green-SCBS-preferred vs minimization of energy consumption vs best-signal-level + transmission rate (greedy).
39 [150]	Long timescale control through use of model predictive control (MPC) based on traffic flow and weather forecast.
40 [150]	RE generation model: Weibull probability distribution to characterize wind speed, with average wind speed obtained from a two-year study of mean wind speed profiles in the Moscow air basin.
41 [150]	RE generation model: Solar radiation data from the Colombian Institute of Hydrology, Meteorology and Environmental Studies (IDEAM).
42 [150]	BS power consumption model: from ICT-EARTH (linear with traffic load specified for time slot t).
43 [150]	Traffic model: Short timescale, inhomogeneous Poisson point process (without temporal variability of traffic) and random-uniform spatial distribution of users in coverage area.
44 [150]	Traffic model: Traffic data provided by a mobile network operator (MNO).
45 [149]	Radio path: path loss calculation using rural microcell and data rate obtained using 5G R16.
46 [149,150]	Simulation, using Matlab.
47 [146]	RE generation model: solar energy production/15 minutes expressed using Gaussian distribution; wind-energy production expressed as constant.

Appendix A.2.3. Table of D-Node Codes

Table A3. Developments.

Development #	Development Code
1 [142]	Enablement of historical analysis of sensor data.
2 [142]	Diesel consumption model for BSs powered by a DG.
3 [142]	Comparison: energy efficiency of BSs powered by a DG with those powered by a RES
4 [142]	Capacity planning of RES power systems for autonomous operation.
5 [142]	CAPEX: PV system < DG system < PV+wind turbine system.
6 [143]	OPEX and energy efficiency of BS in rural/remote areas is significantly improved by using RES.
7 [143]	Autonomy improved, possibly indefinitely, using several RESs.
8 [144]	Optimal, long timescale contract between energy aggregator and CSP, based on availability, during a given time interval of ES discharge capacity at the CSP's BS site
9 [144]	Lyapunov optimization achieving short timescale objective of stable energy levels in ESs and full absorption of energy harvested from PV panels.
10 [144]	Case study, showing near-complete (99.87%) absorption of PV energy.

Table A3. *Cont.*

Development #	Development Code
11 [145]	ML algorithms: correct prediction of traffic only important around utilization threshold that activates/deactivates micro-BS sleep mode.
12 [145]	Comparison: ML algorithms: simple Block Linear Regression performs trade-off between energy consumption and QoS as well as more complex ones such as 24- and 48-Artificial-Neural-Network (ANN).
13 [145]	Energy consumption control strategy: turn micro-BSs on or off according to availability of RE (the sum of the predicted harvest and the stored) has less dependence on ML algorithm than according to demand.
14 [146]	Energy-cost model, integrating TOU grid electricity cost, solar energy and wind energy.
15 [146]	Comparison: algorithms: algorithm minimizing energy cost performs better than one minimizing energy consumption in the overall objective of optimizing energy cost and energy price and energy consumption and fairness of allocation of throughput to users.
16 [147]	Comparison: extensive characterization of the relative merits of four electricity generation systems: (a) PV + ES (b) PV + DG + ES (c) DG + ES (d) DG, in five dimensions: (i) optimal dispatch strategy (ii) NPC (iii) COE (iv) GHG (v) days of autonomy.
17 [148]	Genetic algorithm: minimization of energy cost and wastage in a multiple-RES system.
18 [148]	RES mix recommendation for a specific service area.
19 [149]	Application of green algorithm in 5G HetNet: for reactive, short timescale control, consisting of UA with a green (SC)BS.
20 [149]	Comparison: Green algorithm vs best-signal strength algorithm: UA with grid-powered MBS is reduced almost by half.
21 [149]	Comparison: Green algorithm vs best-signal strength algorithm: Users denied service significantly reduced.
22 [149]	Comparison: Green algorithm vs best-signal strength algorithm: Reduction in MBS energy consumption is limited by the need to backhaul SCBS traffic through the MBS.
23 [149]	Comparison: Green algorithm vs best-signal strength algorithm: Cell site's energy efficiency is improved (MBS's energy efficiency is worsened but SCBSs' improvement over-compensates).
24 [150]	Green algorithm: for reactive, short timescale control of UA with a green (SC)BS.
25 [150]	Comparison: Green algorithm vs on-grid only operation: Energy consumption reduced, and reduction improves markedly when growing from four RE-powered SCBSs (11.1%) to sixteen (26.6%), then to thirty-six (33.7%).
26 [150]	Comparison: Green algorithm vs best-signal-level policy vs best-signal-level + transmission-rate (greedy) heuristic: Energy consumption: significantly better than best-signal-level policy, slightly better than greedy heuristic.
27 [150]	Comparison: Green algorithm vs discrete (MILP) optimizer vs best-signal-level + transmission-rate (greedy) heuristic: Energy efficiency: better than MILP optimizer but roughly same as greedy heuristic.
28 [150]	Infrastructure dimensioning: Green algorithm's reduction of energy consumption does not decrease in direct proportion to number of SCBSs.
29 [150]	Comparison: Green algorithm vs discrete (MILP) optimizer vs best-signal-level policy: Throughput: slightly worse than best-signal-level, slightly better than discrete (MILP) optimizer.

Table A3. *Cont.*

Development #	Development Code
30 [150]	MPC for predictive, long timescale control, using 0- (reactive), 1- and 5-h prediction horizons.
31 [150]	Comparison: MPC vs on-grid only operation: Energy consumption: reduced by 13.1%.
32 [150]	Infrastructure dimensioning: optimal storage capacity can be determined and enables savings up to 22% with respect to on-grid only operation.

Appendix A.3. Clustering the P-, A- and D-Nodes

P3. OBSTACLES TO MULTI-PARTY PARTICIPATION

6. How can participation of communications service providers (CSPs) in renewable energy (RE) harvesting be encouraged?

5. Disparate ownership of RES plant and automatic distribution network (ADN) infrastructure leads to concerns with security and privacy.

P2. EXPLOITAITON OF MULTIPLE RESs

3. How can the limitations of individual RES be overcome?

12. How can we exploit multiple RES to design a RAN that minimizes grid energy consumption while serving at least 95% of users?

4. Spatio-temporal distribution of light intensity and communication demand misaligns energy supply and load.

P4. REAL-TIME CONTROL OBJECTIVES

13. How can user association (UA) be exploited over a short timescale in a hybrid-powered (RES + hydrocarbons) RAN to minimize grid power consumption?

7. How can real-time energy-scheduling be improved to avoid loss of RE, over-discharge of energy storage (ES) and balance between supply and demand?

P1. MID- TO LONG-TERM CONTROL OBJECTIVES

9. Which long-term strategy of operation should be used over prediction intervals: (a) turn off/on micro-BSs to match network demand OR (b) turn off/on micro-BSs to match RE capacity?

8. How well do machine learning (ML) prediction models based on past traffic and RE harvesting support RE use and Quality of Service (QoS)?

14. How can traffic and weather forecasts be exploited over a long timescale to minimize grid power consumption?

10. How can time-of-use electricity pricing, RE production and fairness of allocation of throughput to users be integrated into a modeling scheme?

11. Which system of autonomous power supply, whether renewable or not, optimizes one or more of the following objectives: net present cost (NPC), cost of energy (COE), GHG emissions?

2. How does CAPEX of diesel fuel generators compare with that of RES plant?

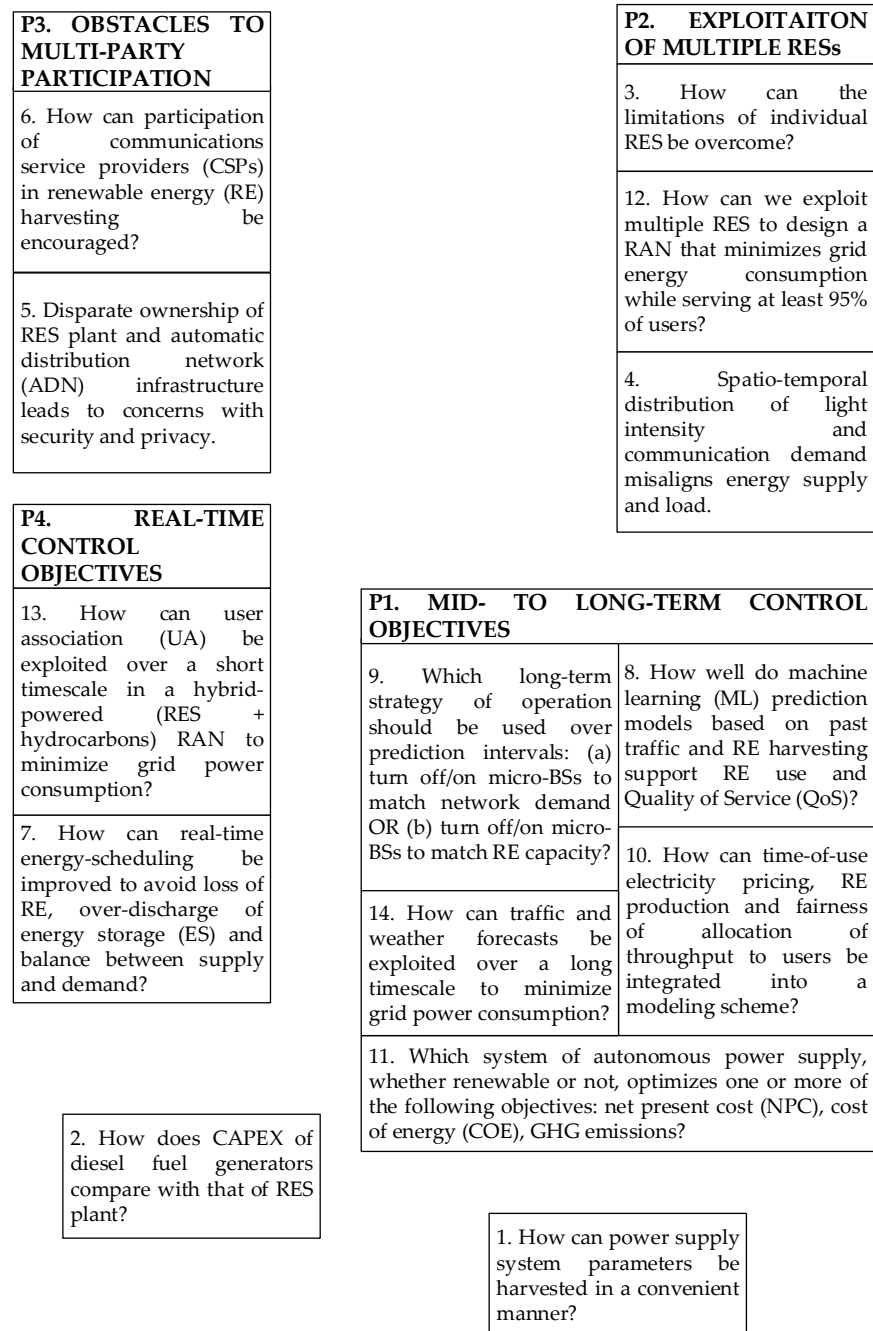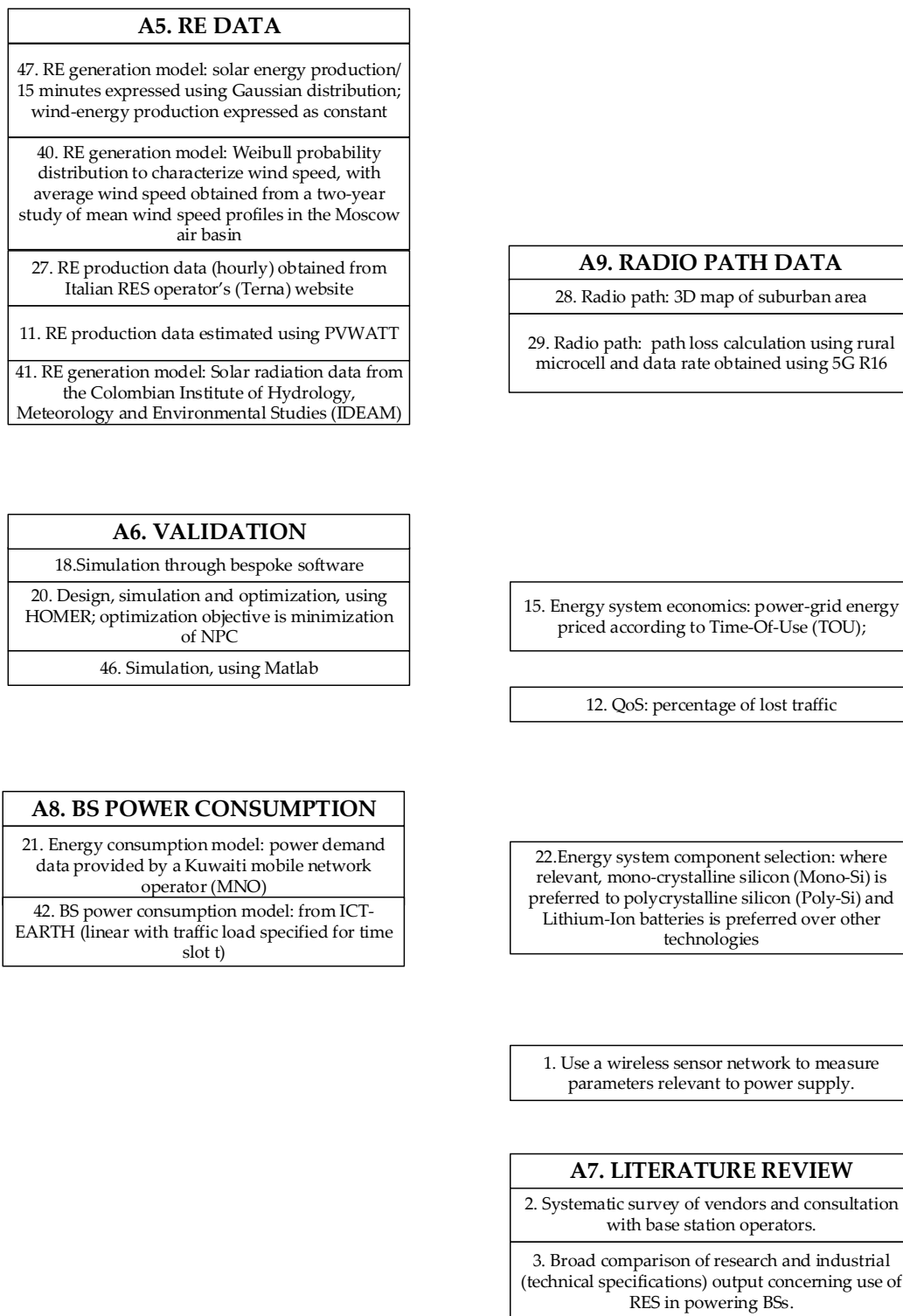1. How can power supply system parameters be harvested in a convenient manner?

Figure A1. A first categorization of the problem nodes. P1: 8 [145], 9 [145], 10 [146], 11 [147], 14 [150]; P2 3 [143], 4 [144], 12 [148]; P3 5 [144], 6 [144]; P4 7 [144], 13 [149,150]. Unclustered: 1 [142], 2 [142].

A1. FORMAL METHODS

39. Long timescale control through use of model predictive control (MPC) based on traffic flow and weather forecast

38. Short timescale control of UA through three alternative policies: green-SCBS-preferred vs minimization of energy consumption vs best-signal-level + transmission rate (greedy)

37. MILP formulation of problem of UA with minimal grid power consumption

17. Alternative energy cost optimization approaches: (a) reduction by decreasing BS energy consumption; (b) reduction through direct optimization of energy cost

23. MILP formulation

29. Genetic algorithm

26. BS power consumption model: linear power model (developed in earlier work), with hourly load factor reflecting days of the week and of the weekend.

6. Regulate the mid-term exchange of energy using contract theory.

7. Regulate the short-term use of energy by a Lyapunov optimization.

16. Game theory applied to balance opposing objectives of (a) minimization of energy cost or energy consumption and (b) maximization of user-throughput fairness

33. Green heuristic: prioritize green SCBSs in the UA process

A2. TRAFFIC MODEL

43. Traffic model: Short timescale, inhomogeneous Poisson point process (without temporal variability of traffic) and random-uniform spatial distribution of users in coverage area

10. Traffic model: Traffic data provided by a large Italian mobile network operator (MNO)

24. Traffic model: Traffic data provided by a mobile network operator (MNO) – both voice and data, for a cell of 0.3km², showing number of simultaneous active users at any time during a day.

32. Random-uniform spatial distribution of low-data-rate and high-data-rate users respectively.

44. Traffic model: Traffic data provided by a mobile network operator (MNO)

A3. RAN ARCHITECTURE

8. RAN architecture: a single service area served by one LTE-A macro-BS (MBS) and six micro-BSs providing additional capacity in the service area's hot spots.

13. RAN architecture: a single service area served by one LTE-A MBS and two micro-BSs providing additional capacity in the service area's hot spots.

25. RAN architecture: eight service areas, each served by one LTE-A MBS and four micro-BSs

30. RAN architecture: a single service area served by one 5G MBS and 24 small cell base stations (SCBSs), with microwave backhaul from MBS to core and SCBS to MBS

34. RAN architecture: a single service area served by one LTE MBS and 36/16/4 small cell base stations (SCBSs)

A4. ENERGY SYSTEM ARCHITECTURE

5. Organize multiple 5G BSs carrying photovoltaic (PV) panels into an energy-aggregation system architecture.

14. Energy system architecture: MBS uses power grid, while one micro-BS is supplied by the power grid and solar energy, and the other micro-BS is supplied by the power grid and wind energy.

9. Energy system architecture: PV panels + energy storage (ES - batteries) + power grid

19. Energy system architecture: Alternative combinations: (a) PV + diesel generator (DG) + ES (b) PV + ES (c) DG + ES (d) DG

31. Energy system architecture: MBS uses power grid, while SCBSs are supplied by RESs.

35. Energy system architecture: MBS uses power grid, while SCBSs are supplied by wind turbines without ES

36. Energy system architecture: MBS uses power grid, while SCBSs are supplied by PV with ES

(a)

Figure A2. *Cont.*

A5. RE DATA

47. RE generation model: solar energy production/ 15 minutes expressed using Gaussian distribution; wind-energy production expressed as constant

40. RE generation model: Weibull probability distribution to characterize wind speed, with average wind speed obtained from a two-year study of mean wind speed profiles in the Moscow air basin

27. RE production data (hourly) obtained from Italian RES operator's (Terna) website

11. RE production data estimated using PVWATT

41. RE generation model: Solar radiation data from the Colombian Institute of Hydrology, Meteorology and Environmental Studies (IDEAM)

A9. RADIO PATH DATA

28. Radio path: 3D map of suburban area

29. Radio path: path loss calculation using rural microcell and data rate obtained using 5G R16

A6. VALIDATION

18. Simulation through bespoke software

20. Design, simulation and optimization, using HOMER; optimization objective is minimization of NPC

46. Simulation, using Matlab

15. Energy system economics: power-grid energy priced according to Time-Of-Use (TOU);

12. QoS: percentage of lost traffic

A8. BS POWER CONSUMPTION

21. Energy consumption model: power demand data provided by a Kuwaiti mobile network operator (MNO)

42. BS power consumption model: from ICT-EARTH (linear with traffic load specified for time slot t)

22. Energy system component selection: where relevant, mono-crystalline silicon (Mono-Si) is preferred to polycrystalline silicon (Poly-Si) and Lithium-Ion batteries is preferred over other technologies

1. Use a wireless sensor network to measure parameters relevant to power supply.

A7. LITERATURE REVIEW

2. Systematic survey of vendors and consultation with base station operators.

3. Broad comparison of research and industrial (technical specifications) output concerning use of RES in powering BSs.

(**b**)

Figure A2. (**a**) A first categorization of the approach nodes. (**b**) A first categorization of the approach nodes. A1: 6 [144], 7 [144], 16 [146], 17 [146], 23 [148], 26 [148], 29 [148], 33 [149], 37 [150], 38 [150], 39 [150]; A2: 10 [145], 24 [148], 32 [149], 43 [150], 44 [150]; A3: 8 [145], 13 [146], 25 [148], 30 [149], 34 [150]; A4: 5 [144], 9 [145], 14 [146], 19 [147], 31 [149], 35 [150], 36 [150]. A5 11 [145], 27 [146], 40 [150], 41 [150], 47 [146]; A6 18 [146], 20 [147], 46 [149,150]; A7 2 [142], 3 [143]; A8 21 [147], 42 [150]; A9 28 [148], 29 [149]. Unclustered: 1 [142], 12 [145], 15 [146], 22 [147].

D1. RELATIVE MERITS OF ALGORITHMS IN ENERGY CONSUMPTION AND EFFICIENCY	
20. Comparison: Green algorithm vs best-signal strength algorithm: UA with grid-powered MBS is reduced almost by half	21. Comparison: Green algorithm vs best-signal strength algorithm: Users denied service significantly reduced
12. Comparison: ML algorithms: simple Block Linear Regression performs trade-off between energy consumption and QoS as well as more complex ones like 24- and 48-Artificial-Neural-Network (ANN)	22. Comparison: Green algorithm vs best-signal strength algorithm: Reduction in MBS energy consumption is limited by the need to backhaul SCBS traffic through the MBS
15. Comparison: algorithms: algorithm minimizing energy cost performs better than one minimizing energy consumption in the overall objective of optimizing energy cost & energy price & energy consumption & fairness of allocation of throughput to users	23. Comparison: Green algorithm vs best-signal strength algorithm: Cell site's energy efficiency is improved (MBS's energy efficiency is worsened but SCBSs' improvement over-compensates).
31. Comparison: MPC vs on-grid only operation: Energy consumption: reduced by 13.1%	25. Comparison: Green algorithm vs on-grid only operation: Energy consumption reduced and reduction improves markedly when growing from 4 RE-powered SCBSs (11.1%) to 16 (26.6%), then to 36 (33.7%)
29. Comparison: Green algorithm vs discrete (MILP) optimizer vs best-signal-level policy: Throughput: slightly worse than best-signal-level, slightly better than discrete (MILP) optimizer	26 Comparison: Green algorithm vs best-signal-level policy vs best-signal-level + transmission-rate (greedy) heuristic: Energy consumption: significantly better than best-signal-level policy, slightly better than greedy heuristic
	27. Comparison: Green algorithm vs discrete (MILP) optimizer vs best-signal-level + transmission-rate (greedy) heuristic: Energy efficiency: better than MILP optimizer but roughly same as greedy heuristic

D2. ALGORITHMS AND OTHER FORMAL SOLUTIONS
8. Optimal, long timescale contract between energy aggregator and CSP, based on availability, during a given time interval of ES discharge capacity at the CSP's BS site.
30. MPC for predictive, long timescale control, using 0-(reactive), 1- and 5-hour prediction horizons
17. Genetic algorithm: minimization of energy cost and wastage in a multiple-RES system
24. Green algorithm: for reactive, short timescale control of UA with a green (SC)BS
9. Lyapunov optimization achieving short timescale objective of stable energy levels in ESs and full absorption of energy harvested from PV panels

D3. AUTONOMOUS OPERATION ON RES(s)
7. Autonomy improved, possibly indefinitely, using several RESs
4. Capacity planning of RES power systems for autonomous operation
18. RES mix recommendation for a specific service area
10. Case study, showing near-complete (99.87%) absorption of PV energy.

(**a**)

Figure A3. *Cont.*

D4. COMPARATIVE ANALYSES OF HYDROCARBON- AND RE-DRIVEN ELECTRICITY SUPPLIES
16. Comparison: extensive characterization of the relative merits of four electricity generation systems: (a) PV+ES (b) PV+DG+ES (c) DG + ES (d) DG, in five dimensions: (i) optimal dispatch strategy (ii) NPC (iii) COE (iv) GHG (v) days of autonomy
3. Comparison: energy efficiency of BSs powered by a DG with those powered by a RES
5. CAPEX: PV system < DG system < PV+wind turbine system

D5. ENERGY COST MODELS
14. Energy-cost model, integrating TOU grid electricity cost, solar energy and wind energy.
2. Diesel consumption model for BSs powered by a DG

1. Enablement of historical analysis of sensor data

D6. INFRASTRUCTURE DIMENSIONING
28. Infrastructure dimensioning: Green algorithm's reduction of energy consumption does not decrease in direct proportion to number of SCBSs.
32. Infrastructure dimensioning: optimal storage capacity can be determined and enables savings up to 22% with respect to on-grid only operation.

6. OPEX and energy efficiency of BS in rural/remote areas is significantly improved by using RES

13. Energy consumption control strategy: turn micro-BSs on or off according to availability of RE (the sum of the predicted harvest & the stored) has less dependence on ML algorithm than according to demand

11. ML algorithms: correct prediction of traffic only important around utilization threshold that activates/ deactivates micro-BS sleep mode

19. Application of green algorithm in 5G HetNet: for reactive, short timescale control, consisting of UA with a green (SC)BS

(**b**)

Figure A3. (**a**) A first categorization of the development nodes. (**b**) A first categorization of the development nodes. D1: 12 [145], 15 [146], 20 [149], 21 [149], 22 [149], 23 [149], 25 [150], 26 [150], 27 [150], 29 [150], 31 [150]; D2: 8 [144], 9 [144], 17 [148], 24 [150], 30 [150]. D3: 4 [142], 7 [143], 10 [144], 18 [148]. D4: 3 [142], 5 [142], 16 [147]; D5: 2 [142], 14 [146]; D6: 28 [150], 32 [150]. Unclustered: 1 [142], 6 [143], 11 [145], 13 [145], 19 [149].

References

1. Depasquale, E.-V.; Salam, H.A.; Davoli, F. PAD: A graphical and numerical enhancement of structural coding to facilitate thematic analysis of a literature corpus. *MethodsX* **2022**, *9*, 101633. [CrossRef]
2. Bolla, R.; Bruschi, R.; Davoli, F.; Cucchietti, F. Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 223–244. [CrossRef]
3. Idzikowski, F.; Chiaraviglio, L.; Cianfrani, A.; Vizcaíno, J.L.; Polverini, M.; Ye, Y. A Survey on Energy-Aware Design and Operation of Core Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1453–1499. [CrossRef]
4. Maaloul, R.; Chaari, L.; Cousin, B. Energy saving in carrier-grade networks: A survey. *Comput. Stand. Interfaces* **2018**, *55*, 8–26. [CrossRef]
5. Delforge, P.; Whitney, J. Data Center Efficiency Assessment—Scaling up Energy Efficiency Across the Data Center Industry: Evaluating Key Drivers and Barriers. Natural Resources Defense Council, IP:14.08-A, August 2014. Available online: https://www.nrdc.org/sites/default/files/data-center-efficiency-assessment-IP.pdf (accessed on 3 October 2020).
6. *Cisco Visual Networking Index: Forecast and Methodology, 2014–2019*; Cisco Systems: San Jose, CA, USA, 2015.
7. *Cisco Visual Networking Index: Forecast and Methodoolgy, 2015–2020*; Cisco Systems: San Jose, CA, USA, 2016.

8. *Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*; Cisco Systems: San Jose, CA, USA, 2017.

9. *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*; Cisco Systems: San Jose, CA, USA, 2019.

10. Ishii, K.; Kurumida, J.; Sato, K.; Kudoh, T.; Namiki, S. Unifying Top-Down and Bottom-Up Approaches to Evaluate Network Energy Consumption. *J. Light. Technol.* **2015**, *33*, 4395–4405. [CrossRef]

11. *Metro Network Traffic Growth: An Architecture Impact Study*; Alcatel-Lucent: Paris, France, 2013.

12. Norton, W.B. *The 2014 Internet Peering Playbook: Connecting to the Core of the Internet*; DrPeering Press: London, UK, 2014.

13. Global Internet Phenomena Report. Sandvine, 2013. Available online: https://www.sandvine.com/hubfs/downloads/archive/2013-2h-global-internet-phenomena-report.pdf (accessed on 4 July 2020).

14. Global Internet Phenomena—Africa, Middle East & North America. December 2015. Available online: https://www.sandvine.com/hubfs/downloads/archive/2015-global-internet-phenomena-report-africa-middle-east-and-north-america.pdf (accessed on 4 July 2020).

15. *Global Internet Phenomena Report*; Sandvine: Plano, TX, USA, 2019.

16. Lee, U.; Rimac, I.; Hilt, V. Greening the internet with content-centric networking. In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, Passau, Germany, 13–15 April 2010; pp. 179–182. [CrossRef]

17. Guan, K.; Atkinson, G.; Kilper, D.C.; Gulsen, E. On the Energy Efficiency of Content Delivery Architectures. In Proceedings of the 2011 IEEE International Conference on Communications Workshops (ICC), Kyoto, Japan, 5–9 June 2011; pp. 1–6. [CrossRef]

18. Choi, N.; Guan, K.; Kilper, D.C.; Atkinson, G. In-network caching effect on optimal energy consumption in content-centric networking. In Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 15 June 2012; pp. 2889–2894. [CrossRef]

19. Osman, N.I.; El-Gorashi, T.; Elmirghani, J.M.H. Reduction of energy consumption of Video-on-Demand services using cache size optimization. In Proceedings of the 2011 Eighth International Conference on Wireless and Optical Communications Networks, Toronto, ON, Canada, 1–4 May 2011; pp. 1–5. [CrossRef]

20. Llorca, J.; Tulino, A.M.; Guan, K.; Esteban, J.; Varvello, M.; Choi, N.; Kilper, D.C. Dynamic in-network caching for energy efficient content delivery. In Proceedings of the 2013 IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 245–249. [CrossRef]

21. Mandal, U.; Chowdhury, P.; Lange, C.; Gladisch, A.; Mukherjee, B. Energy-efficient networking for content distribution over telecom network infrastructure. *Opt. Switch. Netw.* **2013**, *10*, 393–405. [CrossRef]

22. Modrzejewski, R.; Chiaraviglio, L.; Tahiri, I.; Giroire, F.; Le Rouzic, E.; Bonetto, E.; Musumeci, F.; Gonzalez, R.; Guerrero, C. Energy efficient content distribution in an ISP network. In Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, USA, 9–13 December 2013; pp. 2859–2865. [CrossRef]

23. Abji, N.; Tizghadam, A.; Leon-Garcia, A. Energy efficient content delivery in service provider networks with content caching. In Proceedings of the 2015 IEEE Online Conference on Green Communications (OnlineGreenComm), Piscataway, NJ, USA, 10–12 November 2015; pp. 23–29. [CrossRef]

24. Savi, M.; Ayoub, O.; Musumeci, F.; Li, Z.; Verticale, G.; Tornatore, M. Energy-efficient caching for Video-on-Demand in Fixed-Mobile Convergent networks. In Proceedings of the 2015 IEEE Online Conference on Green Communications (OnlineGreenComm), Piscataway, NJ, USA, 10–12 November 2015; pp. 17–22. [CrossRef]

25. Jayasundara, C.; Nirmalathas, A.; Wong, E.; Chan, C. Improving Energy Efficiency of Video on Demand Services. *IEEEOSA J. Opt. Commun. Netw.* **2011**, *3*, 870–880. [CrossRef]

26. Fratini, R.; Savi, M.; Verticale, G.; Tornatore, M. Using replicated video servers for VoD traffic offloading in integrated metro/access networks. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, Australia, 10–14 June 2014; pp. 3438–3443. [CrossRef]

27. Pascale, E.D.; Payne, D.B.; Ruffini, M. Bandwidth and energy savings of locality-aware P2P Content Distribution in next-generation PONs. In Proceedings of the 2012 16th International Conference on Optical Network Design and Modelling (ONDM), Colchester, UK, 17–20 April 2012; pp. 1–6. [CrossRef]

28. Bolla, R.; Bruschi, R.; Davoli, F.; Depasquale, E.-V. Energy-efficient management and control in video distribution networks: 'legacy' hardware based solutions and perspectives of virtualized networking environments. In *Guide to Greening Video Distribution Networks—Energy-Efficient Internet Video Delivery*; Popescu, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2018; pp. 25–57.

29. ITU-T, Y. *3001 Future Networks: Objectives and Design Goals*; International Telecommunication Union: Geneva, Switzerland, 2011.

30. Pervasive Mobile Virtual Services. Expert Advisory Group of the European Technology Platform Networld 2020, July 2016. Available online: https://www.networld2020.eu/wp-content/uploads/2014/02/SRIA_final.pdf (accessed on 23 December 2022).

31. Bouchat, C.; Paul, M.; Allan, D.; Dalle, G. 5G Wireless Wireline Convergence Atchitecture. Broadband Forum, TR-470. August 2020. Available online: https://www.broadband-forum.org/technical/download/TR-470.pdf (accessed on 3 October 2020).

32. *ETSI Standard ETSI ES 203 237 V1.1.1*; Environmental Engineering (EE); Green Abstraction Layer (GAL); Power management Capabilities of the Future Energy Telecommunication Fixed Network Nodes. ETSI: Sophia Antipolis, France, 2014.

33. Environmental Engineering (EE); Green Abstraction Layer (GAL); Power Management Capabilities of The Future Energy Telecommunication Fixed Network Nodes; Enhanced Interface for Power Management in Network Function Virtualisation (NFV) Environments. ETSI, Sophia Antipolis, ETSI ES 203 682 V1.1.0. December 2019. Available online: https://www.etsi.org/deliver/etsi_es/203600_203699/203682/01.01.00_50/es_203682v010100m.pdf (accessed on 31 October 2022).

34. Bolla, R.; Bruschi, R.; Davoli, F.; Lombardo, C.; Pajo, J.F.; Sanchez, O.R. The dark Side of Network Functions Virtualization: A Perspective on the Technological Sustainability. In Proceedings of the IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–7. [CrossRef]
35. *ETSI, RGR/NFV-001ed121*; Network Functions Virtualisation (NFV); Use Cases. ETSI: Sophia Antipolis, France, 2017.
36. Shi, W. Making Sense of the Telco Cloud. Telecoms.com, May 2020. Available online: https://telecoms.com/opinion/making-sense-of-the-telco-cloud/ (accessed on 27 June 2020).
37. CNF Friendly Networking for Telco/Edge Kubernetes Platforms. September 2020. Available online: https://www.brighttalk.com/webcast/12229/438035/cnf-friendly-networking-for-telco-edge-kubernetes-platforms (accessed on 28 September 2020).
38. Crawshaw, J. Deconstructing the Telco Cloud. Light Read. April 2019. Available online: https://www.lightreading.com/cloud/deconstructing-the-telco-cloud/a/d-id/753874 (accessed on 25 July 2020).
39. Ayoubi, S.E.E.; Jeux, S.; Marache, F.; Pujol, F.; Fallgren, M.; Spapis, P.; Yang, C.; Widaa, A.; Markendahl, J.; Ghanbari, A.; et al. Refined Scenarios and Requirements, Consolidated Use Cases, and Qualitative Techno-Economic Feasibility Assessment. D1.1. January 2016. Available online: https://metis-ii.5g-ppp.eu/wp-content/uploads/deliverables/METIS-II_D1.1_v1.0.pdf (accessed on 23 December 2022).
40. Muthurajan, J. Why Use Cloud-Native Containerized Network Functions (CNFs)? Intel, White paper. Available online: https://www.intel.com/content/www/uk/en/communications/why-containers-and-cloud-native-functions-paper.html (accessed on 25 July 2020).
41. Vettor, R.; Smith, S.A. Architecting Cloud Native .NET Applications for Azure. October 2020. Available online: https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/ (accessed on 30 October 2022).
42. Logan, M. CNCF F Technical Oversight Committee (TOOC). GitHub, March 2021. Available online: https://github.com/cncf/toc (accessed on 30 October 2022).
43. DPDK. DPDK, 2021. Available online: https://www.dpdk.org/ (accessed on 25 February 2021).
44. Dayarathna, M.; Wen, Y.; Fan, R. Data Center Energy Consumption Modeling: A Survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 732–794. [CrossRef]
45. Costa, G.D.; Pierson, J.-M.; Fontoura-Cupertino, L. Effectiveness of Neural Networks for Power Modeling for Cloud and HPC: It's Worth It! *ACM Trans. Model. Perform. Eval. Comput. Syst.* **2020**, *5*, 1–36. [CrossRef]
46. Ismail, L.; Materwala, H. Computing Server Power Modeling in a Data Center: Survey, Taxonomy, and Performance Evaluation. *ACM Comput. Surv.* **2020**, *53*, 1–34. [CrossRef]
47. Lin, W.; Shi, F.; Wu, W.; Li, K.; Wu, G.; Mohammed, A.-A. A Taxonomy and Survey of Power Models and Power Modeling for Cloud Servers. *ACM Comput. Surv.* **2020**, *53*, 100:1–100:41. [CrossRef]
48. Fu, S.; Liu, J.; Zhu, W. Multimedia Content Delivery with Network Function Virtualization: The Energy Perspective. *IEEE Multimed.* **2017**, *24*, 38–47. [CrossRef]
49. Chen, Q.; Grosso, P.; van der Veldt, K.; de Laat, C.; Hofman, R.; Bal, H. Profiling Energy Consumption of VMs for Green Cloud Computing. In Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, Sydney, Australia, 12–14 December 2011; pp. 768–775. [CrossRef]
50. Khan, N. Investigating Energy Efficiency of Physical and Virtual Machines in Cloud Computing. Univeristy of Oslo, 2017. Available online: https://www.duo.uio.no/handle/10852/61321 (accessed on 7 November 2019).
51. Li, G.; Zhang, D.; Li, Y.; Li, K. Toward energy-efficiency optimization of pktgen-DPDK for green network testbeds. *China Commun.* **2018**, *15*, 199–207. [CrossRef]
52. Intel®Ethernet Server Adapter I350: Product Brief. Intel, March 2021. Available online: https://www.intel.com/content/www/us/en/products/docs/network-io/ethernet/10-25-40-gigabit-adapters/ethernet-i350-server-adapter-brief.html (accessed on 9 February 2021).
53. Xu, Z.; Liu, F.; Wang, T.; Xu, H. Demystifying the energy efficiency of Network Function Virtualization. In Proceedings of the 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), Beijing, China, 20–21 June 2016; pp. 1–10. [CrossRef]
54. Zhang, Z.; Fu, S. Macropower: A coarse-grain power profiling framework for energy-efficient cloud computing. In Proceedings of the 30th IEEE International Performance Computing and Communications Conference, Orlando, FL, USA, 17–19 November 2011; pp. 1–8. [CrossRef]
55. Morabito, R. Power Consumption of Virtualization Technologies: An Empirical Investigation. In Proceedings of the 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), Limassol, Cyprus, 7–10 December 2015; pp. 522–527. [CrossRef]
56. Shea, R.; Wang, H.; Liu, J. Power consumption of virtual machines with network transactions: Measurement and improvements. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 1051–1059. [CrossRef]
57. Aldossary, M.; Djemame, K.; Alzamil, I.; Kostopoulos, A.; Dimakis, A.; Agiatzidou, E. Energy-aware cost prediction and pricing of virtual machines in cloud computing environments. *Future Gener. Comput. Syst.* **2019**, *93*, 442–459. [CrossRef]
58. Li, Y.; Wang, Y.; Yin, B.; Guan, L. An Online Power Metering Model for Cloud Environment. In Proceedings of the 2012 IEEE 11th International Symposium on Network Computing and Applications, Cambridge, MA, USA, 23–25 August 2012; pp. 175–180. [CrossRef]

59. Wen, C.; Long, X.; Yang, Y.; Ni, F.; Mu, Y. System Power Model and Virtual Machine Power Metering for Cloud Computing Pricing. In Proceedings of the 2013 Third International Conference on Intelligent System Design and Engineering Applications, Hong Kong, 16–18 January 2013; pp. 1379–1382. [CrossRef]

60. Arroba, P.; Risco-Martín, J.L.; Zapater, M.; Moya, J.M.; Ayala, J.L.; Olcoz, K. Server Power Modeling for Run-time Energy Optimization of Cloud Computing Facilities. *Energy Procedia* **2014**, *62*, 401–410. [CrossRef]

61. Wu, W.; Lin, W.; Peng, Z. An intelligent power consumption model for virtual machines under CPU-intensive workload in cloud environment. *Soft Comput.* **2017**, *21*, 5755–5764. [CrossRef]

62. Dhiman, G.; Mihic, K.; Rosing, A. T system for online power prediction in virtualized environments using Gaussian mixture models. In Proceedings of the 47th Design Automation Conference, Anaheim, CA, USA, 13–18 June 2010; p. 807. [CrossRef]

63. Salam, H.A.; Davoli, F.; Timm-Giel, A. Improving Prediction Accuracy for Power Consumption in Virtual Environments. In Proceedings of the 2019 29th International Telecommunication Networks and Applications Conference (ITNAC), Auckland, New Zealand, 27–29 November 2019; pp. 1–6. [CrossRef]

64. Salam, H.A.; Davoli, F.; Carrega, A.; Timm-Giel, A. Towards Prediction of Power Consumption of Virtual Machines for Varying Loads. In Proceedings of the 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, Australia, 21–23 November 2018; pp. 1–6. [CrossRef]

65. MCallau-Zori; Samoila, L.; Orgerie, A.-C.; Pierre, G. An experiment-driven energy consumption model for virtual machine management systems. *Sustain. Comput. Inform. Syst.* **2018**, *18*, 163–174. [CrossRef]

66. Chinprasertsuk, S.; Gertphol, S. Power model for virtual machine in cloud computing. In Proceedings of the 2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE), Chonburi, Thailand, 14–16 May 2014; pp. 140–145. [CrossRef]

67. Enokido, T.; Takizawa, M. The Extended Power Consumption Model to Perform Computation Type Application Processes on Virtual Machines. In Proceedings of the 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS), Fukuoka, Japan, 6–8 July 2016; pp. 15–22. [CrossRef]

68. Enokido, T.; Takizawa, M. Power Consumption Model of a Server to Perform Communication Type Application Processes on Virtual Machines. In Proceedings of the 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), Krakow, Poland, 4–6 November 2015; pp. 275–282. [CrossRef]

69. Krishnan, B.; Amur, H.; Gavrilovska, A.; Schwan, K. VM power metering: Feasibility and challenges. *ACM SIGMETRICS Perform. Eval. Rev.* **2011**, *38*, 56. [CrossRef]

70. Isci, C.; Martonosi, M. Runtime power monitoring in high-end processors: Methodology and empirical data. In Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36, Wahington, DC, USA, 3–6 December 2003; pp. 93–104. [CrossRef]

71. Enokido, T.; Takizawa, M. Power Consumption and Computation Models of Virtual Machines to Perform Computation Type Application Processes. In Proceedings of the 2015 Ninth International Conference on Complex, Intelligent, and Software Intensive Systems, Santa Catarina, Brazil, 8–10 July 2015; pp. 126–133. [CrossRef]

72. Aldossary, M.; Djemame, K. Performance and Energy-based Cost Prediction of Virtual Machines Live Migration in Clouds. In Proceedings of the 8th International Conference on Cloud Computing and Services Science, Funchal, Madeira, Portugal, 19–21 March 2018; pp. 384–391. [CrossRef]

73. Aldossary, M.; Djemame, K. Energy-based Cost Model of Virtual Machines in a Cloud Environment. In Proceedings of the 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), Amman, Jordan, 31 October 2018; pp. 1–8. [CrossRef]

74. Alzamil, I.; Djemame, K. Energy Prediction for Cloud Workload Patterns. In *Economics of Grids, Clouds, Systems, and Services*; Springer: Cham, Switzerland, 2017; pp. 160–174. [CrossRef]

75. Kansal, A.; Zhao, F.; Liu, J.; Kothari, N.; Bhattacharya, A.A. Virtual machine power metering and provisioning. In Proceedings of the 1st ACM Symposium on Cloud Computing—SoCC '10, Indianapolis, IN, USA, 10–11 June 2010; p. 39. [CrossRef]

76. Waßmann, I.; Versick, D.; Tavangarian, D. Energy consumption estimation of virtual machines. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; pp. 1151–1156. [CrossRef]

77. Gu, C.; Shi, P.; Shi, S.; Huang, H.; Jia, X. A Tree Regression Based Approach for VM Power Metering. *IEEE Access* **2015**, *3*, 610–621. [CrossRef]

78. Silva-de-Souza, W.; Iranfar, A.; Bráulio, A.; Zapater, M.; Xavier-de-Souza, S.; Olcoz, K.; Atienza, D. Containergy—A Container-Based Energy and Performance Profiling Tool for Next Generation Workloads. *Energies* **2020**, *13*, 2162. [CrossRef]

79. Veni, T.; Bhanu, S.M.S. Prediction Model for Virtual Machine Power Consumption in Cloud Environments. *Procedia Comput. Sci.* **2016**, *87*, 122–127. [CrossRef]

80. SPEC Benchmarks. December 2020. Available online: https://www.spec.org/benchmarks.html (accessed on 30 October 2022).

81. Phung, J.; Young, C.L.; Zomaya, A.Y. Application-Agnostic Power Monitoring in Virtualized Environments. In Proceedings of the 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Madrid, Spain, 14–17 May 2017; pp. 335–344. [CrossRef]

82. Phung, J.; Lee, Y.C.; Zomaya, A.Y. Lightweight Power Monitoring Framework for Virtualized Computing Environments. *IEEE Trans. Comput.* **2020**, *69*, 14–25. [CrossRef]

83. Bolla, R.; Bruschi, R.; Davoli, F.; Pajo, J.F. A Model-Based Approach Towards Real-Time Analytics in NFV Infrastructures. *IEEE Trans. Green Commun. Netw.* **2020**, *4*, 529–541. [CrossRef]

84. Bohra, A.E.H.; Chaudhary, V. VMeter: Power modelling for virtualized clouds. In Proceedings of the 2010 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), Atlanta, GA, USA, 19–23 April 2010; pp. 1–8. [CrossRef]

85. Zorello, L.M.M.; Vieira, M.G.T.; Tejos, R.A.G.; Rojas, M.A.T.; Meirosu, C.; de Brito Carvalho, T.C.M. Improving Energy Efficiency in NFV Clouds with Machine Learning. In Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2–7 July 2018; pp. 710–717. [CrossRef]

86. Stoess, J.; Lang, C.; Bellosa, F. Energy Management for Hypervisor-Based Virtual Machines. In Proceedings of the USENIX Annual Technical Conference, Santa Clara, CA, USA, 17–22 June 2007.

87. Li, X.; Cheng, W.; Zhang, T.; Ren, F.; Yang, B. Towards Power Efficient High Performance Packet I/O. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 981–996. [CrossRef]

88. Virtio—KVM. 2020. Available online: https://www.linux-kvm.org/page/Virtio (accessed on 27 August 2020).

89. Poll Mode Driver—Data Plane Development Kit 20.11.0-rc0 Documentation. December 2014. Available online: https://doc.dpdk.org/guides/prog_guide/poll_mode_drv.html (accessed on 24 December 2022).

90. The Zeek Network Security Monitor. Zeek, March 2020. Available online: https://zeek.org/ (accessed on 2 October 2020).

91. Snort—Network Intrusion Detection & Prevention System. May 2015. Available online: https://www.snort.org/ (accessed on 2 October 2020).

92. Open vSwitch. 2020. Available online: https://www.openvswitch.org/ (accessed on 27 August 2020).

93. Rizzo, L.; Lettieri, G. VALE, a switched ethernet for virtual machines. In Proceedings of the Proceedings of the 8th international conference on Emerging networking experiments and Technologies, New York, NY, USA, 6–9 December 2012; pp. 61–72. [CrossRef]

94. Bruschi, R.; Davoli, F.; Lago, P.; Pajo, J.F. Joint Power Scaling of Processing Resources and Consolidation of Virtual Network Functions. In Proceedings of the 5th IEEE International Conference on Cloud Networking (Cloudnet), Pisa, Italy, 3–6 October 2016; pp. 70–75. [CrossRef]

95. Wu, W.; Lin, W.; He, L.; Wu, G.; Hsu, C. A Power Consumption Model for Cloud Servers Based on Elman Neural Network. *IEEE Trans. Cloud Comput.* **2019**, *9*, 1268–1277. [CrossRef]

96. Łukasik, R. Run-to-completion model in Data Plane processing—Nokia Wrocław. December 2019. Available online: https://nokiawroclaw.pl/blog/post/run-to-completion-model-in-data-plane-processing/ (accessed on 30 September 2020).

97. Kai, C.; Blesson, V.; Peter, K.; Nikolopoulos, D.S. Power Modelling for Heterogeneous Cloud-Edge Data Centers. *Adv. Parallel Comput.* **2018**, *32*, 804–813. [CrossRef]

98. Khan, K.N.; Hirki, M.; Niemi, T.; Nurminen, J.K.; Ou, Z. RAPL in Action: Experiences in Using RAPL for Power Measurements. *ACM Trans. Model. Perform. Eval. Comput. Syst.* **2018**, *3*, 9:1–9:26. [CrossRef]

99. Molnar, I. Performance Counters for Linux, v8. 2009. Available online: https://lwn.net/Articles/336542/ (accessed on 23 July 2020).

100. Bolla, R.; Bruschi, R.; Carrega, A.; Davoli, F. Green Networking with Packet Processing Engines: Modeling and Optimization. *IEEE ACM Trans. Netw.* **2014**, *22*, 110–123, Correction in *IEEE ACM Trans. Netw.* **2017**, *22*, 110–123. [CrossRef]

101. TPC-W Homepage. 2021. Available online: http://www.tpc.org/tpcw/ (accessed on 11 February 2021).

102. Advanced Configuration and Power Interface (ACPI) Specification—ACPI Specification 6.4 Documentation. January 2021. Available online: https://uefi.org/specs/ACPI/6.4/ (accessed on 2 March 2021).

103. Keong, C.K.; Wei, K.T.; Ghani, A.A.A.; Sharif, K.Y. Toward using software metrics as indicator to measure power consumption of mobile application: A case study. In Proceedings of the 2015 9th Malaysian Software Engineering Conference (MySEC), Kuala Lumpur, Malaysia, 16–17 December 2015; pp. 172–177. [CrossRef]

104. Li, K. Optimal configuration of a multicore server processor for managing the power and performance tradeoff. *J. Supercomput.* **2012**, *61*, 189–214. [CrossRef]

105. Mobius, C.; Dargie, W.; Schill, A. Power Consumption Estimation Models for Processors, Virtual Machines, and Servers. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 1600–1614. [CrossRef]

106. Technical Committee ISO/TC 207/SC 5 Life Cycle Assessment—Principles and Framework. Standard ISO 14040:2006. July 2006. Available online: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/74/37456.html (accessed on 27 July 2020).

107. L.1310: Energy Efficiency Metrics and Measurement Methods for Telecommunication Equipment. July 2017. Available online: https://www.itu.int/rec/T-REC-L.1310/en (accessed on 27 July 2020).

108. Schien, D.; Shabajee, P.; Yearworth, M.; Preist, C. Modeling and Assessing Variability in Energy Consumption During the Use Stage of Online Multimedia Services. *J. Ind. Ecol.* **2013**, *17*, 800–813. [CrossRef]

109. Measurement Method for Energy Efficiency of Network Functions Virtualization. November 2018. Available online: https://www.itu.int/rec/T-REC-L.1361/en (accessed on 23 July 2020).

110. Hennessy, J.L.; Patterson, D.A. *Computer Architecture: A Quantitative Approach*; Morgan Kaufmann: Burlington, MA, USA, 2017.

111. Inoue, T.; Aikebaier, A.; Enokido, T.; Takizawa, M. Algorithms for Selecting Energy-Efficient Storage Servers in Storage and Computation Oriented Applications. In Proceedings of the 2012 IEEE 26th International Conference on Advanced Information Networking and Applications, Fukuoka, Japan, 26–29 March 2012; pp. 920–927. [CrossRef]

112. Al-Zubaedi, W.; Al-Raweshidy, H.S. A parameterized and optimized BBU pool virtualization power model for C-RAN architecture. In Proceedings of the IEEE EUROCON 2017—17th International Conference on Smart Technologies, Ohrid, Macedonia, 6–8 July 2017; pp. 38–43. [CrossRef]

113. Mijumbi, R.; Serrat, J.; Gorricho, J.-L.; Rubio-Loyola, J. On the Energy Efficiency Prospects of Network Function Virtualization. ArXiv151200215 Cs. December 2015. Available online: http://arxiv.org/abs/1512.00215 (accessed on 15 July 2020).

114. Dlamini, T. Softwarization in Future Mobile Networks and Energy Efficient Networks. In *Mobile Computing*; Intech Open: London, UK, 2019; p. 51. [CrossRef]

115. Dumitrescu, C. Design Patterns for Packet Procesing Applications on Multi-core Intel Architecture. December 2008. Available online: https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/ia-multicore-packet-processing-paper.pdf (accessed on 29 September 2020).

116. Hähnel, M.; Döbel, B.; Völp, M.; Härtig, H. Measuring energy consumption for short code paths using RAPL. *ACM SIGMETRICS Perform. Eval. Rev.* **2012**, *40*, 13–17. [CrossRef]

117. TelecomTV The 5G Core is Vital to Deliver the Promise of 5G. TelecomTV, July 2020. Available online: https://www.telecomtv.com/content/intel-vsummit-5g-ran-5g-core/the-5g-core-is-vital-to-deliver-the-promise-of-5g-39164/ (accessed on 17 February 2021).

118. Bhyrraju, R.; Druta, D.; Brenner, M.; Gautam, D.; Kitroser, I.; Wocalewski, T.; Wright, S.; Feng, A.; Shao, G.; Liu, J.; et al. 'VNF Descriptor and Packaging Specification', ETSI, Sophia Antipolis Cedex-FRANCE, ETSI GS NFV-IFA 011. September 2019. Available online: https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/011/02.07.01_60/gs_NFV-IFA011v020701p.pdf (accessed on 24 December 2022).

119. The NANOG February 2021 Archive by Thread. NANOG Mailing List Archive. 2021. Available online: https://mailman.nanog.org/pipermail/nanog/2021-February/thread.html (accessed on 5 March 2021).

120. Bolla, R.; Bruschi, R.; Carrega, A.; Davoli, F. Green network technologies and the art of trading-off. In Proceedings of the 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Shanghai, China, 10–15 April 2011; pp. 301–306. [CrossRef]

121. Dargie, W. A Stochastic Model for Estimating the Power Consumption of a Processor. *IEEE Trans. Comput.* **2015**, *64*, 1311–1322. [CrossRef]

122. Garraghan, P.; Al-Anii, Y.; Summers, J.; Thompson, H.; Kapur, N.; Djemame, K. A Unified Model for Holistic Power Usage in Cloud Datacenter Servers. In Proceedings of the 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC), Shanghai, China, 6–9 December 2016; pp. 11–19.

123. Monteiro, A.F.; Loques, O. Quantum Virtual Machine: A Scalable Model to Optimize Energy Savings and Resource Management. In Proceedings of the 2015 27th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), Florianopolis, Brazil, 17–21 October 2015; pp. 194–201. [CrossRef]

124. Sharma, S.; Chang, V.; Tim, U.S.; Wong, J.L.H.; Gadia, S.K. Cloud-based emerging services systems. *Int. J. Inf. Manag.* **2016**, *640*, 71–91. [CrossRef]

125. Zakarya, M.; Gillam, L. An Energy Aware Cost Recovery Approach for Virtual Machine Migration. In *Economics of Grids, Clouds, Systems, and Services*; Springer: Cham, Switzerland, 2017; pp. 175–190. [CrossRef]

126. Zhao, T.; Wu, J.; Zhou, S.; Niu, Z. Energy-delay tradeoffs of virtual base stations with a computational-resource-aware energy consumption model. In Proceedings of the 2014 IEEE International Conference on Communication Systems, San Jose, CA, USA, 3–6 November 2014; pp. 26–30. [CrossRef]

127. Marletta, A. CPU Usage Limiter for Linux. 2012. Available online: http://cpulimit.sourceforge.net/ (accessed on 27 July 2020).

128. Son, J.; He, T.; Buyya, R. CloudSimSDN-NFV: Modeling and simulation of network function virtualization and service function chaining in edge computing environments. *Softw. Pract. Exp.* **2019**, *49*, 1748–1764. [CrossRef]

129. Vasan, A.; Sivasubramaniam, A.; Shimpi, V.; Sivabalan, T.; Subbiah, R. Worth their watts? An empirical study of datacenter servers. In Proceedings of the HPCA —16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture, Bangalore, India, 9–14 January 2010; pp. 1–10. [CrossRef]

130. Slomon, D.A.; Yosifovich, P.; Russinovich, M.E.; Lonescu, A. System Architecture, Processes, Threads, Memory Management, and More, 7th ed. Pearson Education, 2017. Available online: https://www.oreilly.com/library/view/windows-internals-seventh/9780133986471/ (accessed on 27 July 2020).

131. PU Analysis. Micrsoft, May 2017. Available online: https://docs.microsoft.com/en-us/windows-hardware/test/wpt/cpu-analysis (accessed on 27 July 2020).

132. Goraczko, M. Joulemeter: Computational Energy Measurement and Optimization. Microsoft Research, February 2010. Available online: https://www.microsoft.com/en-us/research/project/joulemeter-computational-energy-measurement-and-optimization/ (accessed on 27 July 2020).

133. Lin, W.; Wu, W.; Wang, H.; Wang, J.Z.; Hsu, C.-H. Experimental and quantitative analysis of server power model for cloud data centers. *Future Gener. Comput. Syst.* **2018**, *86*, 940–950. [CrossRef]

134. Aldossary, M.; Alzamil, I.; Djemame, K. Towards Virtual Machine Energy-Aware Cost Prediction in Clouds. In *Economics of Grids, Clouds, Systems, and Services*; Springer: Cham, Switzerland, 2017; pp. 119–131. [CrossRef]

135. Intel®Core^TM i5-3230M Processor (3M Cache, up to 3.20 GHz) rPGA Product Specifications. May 2014. Available online: https://ark.intel.com/content/www/us/en/ark/products/72164/intel-core-i5-3230m-processor-3m-cache-up-to-3-20-ghz-rpga.html (accessed on 27 July 2020).

136. Mobile 3rd Generation Intel®CoreTM Processor Family, Mobile Intel® Pentium® Processor Family and Mobile Intel® Celeron® Processor Family. Datasheet 326768–006, June 2013. Available online: https://www.intel.com/content/dam/www/public/us/en/documents/datasheets/3rd-gen-core-family-mobile-vol-1-datasheet.pdf (accessed on 27 July 2020).

137. Doleželová, M.; Skarvada, J.; Heves, J.; Ruseva, Y.; Reed, J.; Landmann, R.; Domingo, D. Power Management Guide Red Hat Enterprise Linux 6. *Red Hat Customer Portal*. December 2010. Available online: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/power_management_guide/index (accessed on 24 December 2022).

138. Chios, M. Network Functions Virtualisation. An Introduction, Benefits, Enablers, Challenges & Call for Action. In Proceedings of the SDN and OpenFlow World Congress, Darmstadt, Germany, 15–18 October 2012.

139. Molka, D.; Hackenberg, D.; Schöne, R.; Müller, M.S. Characterizing the energy consumption of data transfers and arithmetic operations on x86-64 processors. In Proceedings of the International Conference on Green Computing, Chicago, IL, USA, 15–18 August 2010; pp. 123–133. [CrossRef]

140. Rahman, A.; Lynch, P.; Aranda, P.; Bernardos, C.; Zuniga, J.; Contreras, L. Network Virtualization Research Challenges. Request For Comments (RFC)—Informational RFC 8568, ISSN: 2070–1721. April 2019. Available online: https://tools.ietf.org/html/rfc8568$\backslash$#page-18 (accessed on 10 July 2020).

141. Dahyun SR-IOV for NFV Solutions—Practical Considerations and Thoughts. October 2017. Available online: http://man-glove.blogspot.com/2017/10/sr-iov-for-nfv-solutions-practical.html (accessed on 27 July 2020).

142. Lorincz, J.; Bule, I.; Kapov, M. Performance Analyses of Renewable and Fuel Power Supply Systems for Different Base Station Sites. *Energies* **2014**, *7*, 7816. [CrossRef]

143. Lorincz, J.; Bule, I. Renewable Energy Sources for Power Supply of Base Station Sites. *Int. J. Bus. Data Commun. Netw. IJBDCN* **2013**, *9*, 53–74. [CrossRef]

144. Zhang, X.; Wang, Z.; Zhou, Z.; Liao, H.; Ma, X.; Yin, X.; Lv, G.; Wang, Z.; Lu, Z.; Liu, Y. Optimal Dispatch of Multiple Photovoltaic Integrated 5G Base Stations for Active Distribution Network Demand Response. *Front. Energy Res.* **2022**, *10*, 787. Available online: https://www.frontiersin.org/articles/10.3389/fenrg.2022.919197 (accessed on 17 December 2022). [CrossRef]

145. Vallero, G.; Renga, D.; Meo, M.; Marsan, M.A. Greener RAN Operation Through Machine Learning. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 896–908. [CrossRef]

146. Guo, W.; Shin, D.R.; Siddiqui, I.F.; Koo, J.; Qureshi, N.M.F. Cost-Effective Optimal Multi-Source Energy Management Technique in Heterogeneous Networks. In Proceedings of the 2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW), Bologna, Italy, 13 July 2022; pp. 181–186. [CrossRef]

147. Baidas, M.W.; Hasaneya, R.W.; Kamel, R.M.; Alanzi, S.S. Solar-Powered Cellular Base Stations in Kuwait: A Case Study. *Energies* **2021**, *14*, 7494. [CrossRef]

148. Deruyck, M.; Bova, S.; Vallero, G.; Meo, M.; Martens, L.; Joseph, W. Designing a hybrid renewable energy source system to feed the wireless access network. *Sustain. Energy Grids Netw.* **2022**, *31*, 100722. [CrossRef]

149. Samorzewski, A.; Kliks, A. User Allocation in Heterogeneous Network Supplied by Renewable Energy Sources. In Proceedings of the 2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Bologna, Italy, 11–13 October 2021; pp. 419–422. [CrossRef]

150. Fletscher, L.A.; Suárez, L.A.; Grace, D.; Peroni, C.V.; Maestre, J.M. Energy-Aware Resource Management in Heterogeneous Cellular Networks with Hybrid Energy Sources. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 279–293. [CrossRef]